

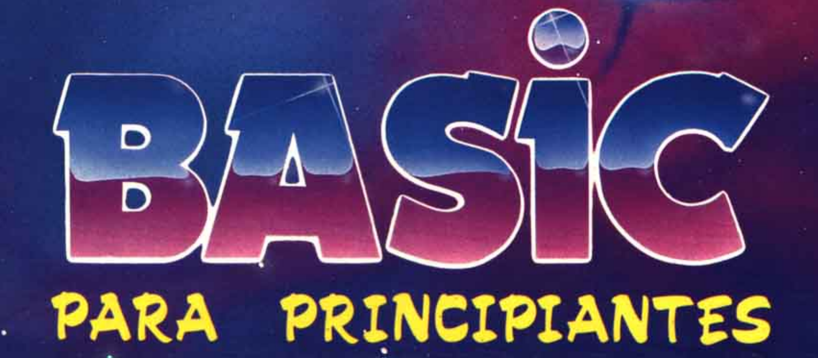
AÑO I
Núm. 11
Marzo
1986
300 Ptas.

Logo, un lenguaje educativo

Screen 3, el modo multicolor

Aplicación: Sistemas de ecuaciones

Libros, Software, Programas...



EDITORIAL





El BASIC ha sido el lenguaje de programación por excelencia de los ordenadores personales. Sin embargo, apenas hemos hecho alusión al tema a lo largo de los meses. Por esta razón y a partir del número que se encuentra en tus manos, empezamos una serie que tratará los aspectos del BASIC MSX, un lenguaje con unas 140 instrucciones de las cuales se utilizan las 2/3 partes de ellas. Profundizaremos, tanto en los aspectos del lenguaje como en sus características más importantes, desde las instrucciones de control hasta las más complicadas de tratamientos de ficheros, ya sean estos en cinta o en disco.

Otro lenguaje de programación, que está teniendo un auge sin precedentes es el LOGO, tema de portada de este mes. El artículo, basado en el MS-LOGO de Philips, es una breve historia de este lenguaje de programación y de cómo empezó en los colegios. A lo largo de todo el artículo se muestran una serie de pantallas como ejemplos prácticos de este importante lenguaje, cuyas aplicaciones abarcan todo tipo de actividades.

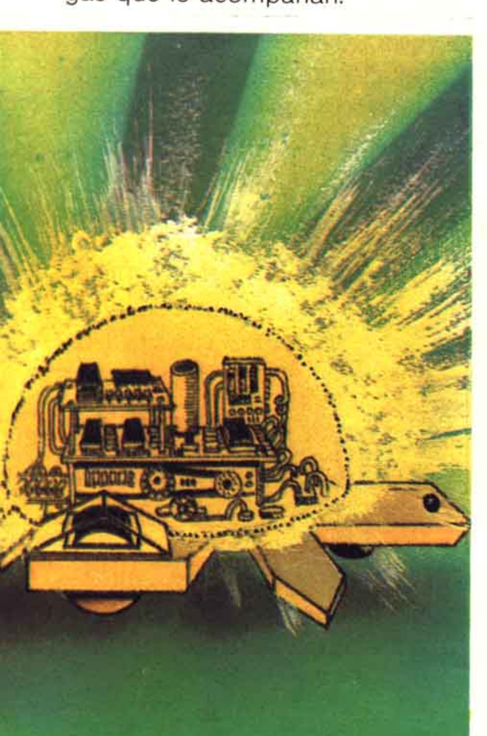
Por otro lado, continuamos con las aplicaciones matemáticas y los modos de pantallas, donde este mes dedicaremos especial atención a los sprites y prepararemos el camino para que el usuario se vaya iniciando en la creación de figuras y cómo desplazarlas.

Sin embargo, para el mes que viene tenemos una serie de sorpresas. Por un lado vuelve a nuestras páginas el Código Máquina (con otra forma y otro estilo), ganador indiscutible de aquella encuesta (cuyos premiados aparecerán en el número de abril o mayo en su defecto), de la que hemos podido obtener datos muy interesantes. En esta ocasión, la serie tratará más ejemplos y menos teoría. Con esto no queremos decir que sólo nos dedicaremos a publicar programas, nada más lejos de la realidad, sino que los programas van a ser una constante dentro de los artículos.

Y por otra parte, los usuarios del SVI-328 están de enhorabuena. Durante cinco números dedicaremos especial atención al precedesor del SVI-728, que ha estado abandonado durante mucho tiempo.



SUMARIO






OWNER'S MAN
SP-1000
MATRIX PRIN
SEIKOSHA



NOTICIAS

La cámara que se puede conectar a un ordenador

Aunque suene algo extraño, existe una cámara que se puede conectar a un MSX.

Efectivamente, la máquina en cuestión, es la T90 de Canon, la cual, mediante el interface DMB se podrá conectar a cualquier ordenador MSX.

Este aparato incorpora una serie de innovaciones, importantes para el fotógrafo aficionado o para el profesional, que es a quien va dirigida. Es un paso más hacia la fotografía electrónica. Contiene todos los alicientes que se debe esperar de un aparato de esta clase, tales como enfoque automático, ocho modos de fotografía, etc., además de ser la más rápida del mercado, con velocidades de obturados de hasta 1/4.000 de seg.

Otra de las innovaciones de esta cámara consiste en los tres pequeños motores que posee en lugar de uno sólo. Cada uno de ellos cuenta con una función específica como es el avance del carrete, la velocidad del obturador y el rebobinado automático, todo esto está controlado por dos microprocesadores que se comparten la tarea de controlar la secuencia, mostrar la información de la exposición y el control del motor.

Empleando el interface DMB, es posible introducir los datos necesarios acerca de la exposición, fecha y el número de referencia, lo que le convierte en una herramienta muy útil en lugares como consultas médicas, dentistas, etc.

Nuevo ordenador de Toshiba

El modelo HX-10 ya tiene sustitutos dentro de la misma casa, el HX-20 y el HX-22. Los nuevos modelos (el mes que viene comentaremos en la sección de TEST el HX-20) vienen a ocupar una escala intermedia entre los ordenadores básicos y la segunda generación. Estos ordenadores presentan varias innovaciones importantes, entre las que podemos destacar el potente tratamiento de textos que incorpora y posibilidad de trabajar con un disco RAM (este término lo veremos el mes que viene).

Básicamente, es como el modelo anterior, el HX-10, pero el diseño y diversas mejoras, le dan un aspecto más serio que aquel a un precio que se situará alrededor de las 51.000 ptas. (el HX-20 sin I.V.A.).

El modelo HX-22 se diferencia del primero en que, además de incorporar el tratamiento de textos, trae a su vez el interface RS-232C. Sin embargo, el interface también se puede adquirir a parte, a un precio que rondará las 17.000 ptas., mientras que el cable saldrá por unas 7.500 ptas. Para obtener más información, dirigirse a:

E.M.S.A.
Caballero, 79.
08014 Barcelona.

Idealogic: Software para la 2.ª Generación de MSX

TELARIUM es la colección de libros de aventuras para ordenador de más éxito en EE.UU. Las aventuras de TELARIUM disponen de un potente parser (analizador sintáctico de lenguaje natural) capaz de comprender lenguaje escrito y con un vocabulario de alrededor de 500 palabras.

El usuario se convierte en protagonista de la aventura y dispone de pantallas gráficas combinadas con texto para participar de modo directo. En estos títulos la música, los efectos especiales y los gráficos son fundamentales, acercándonos por ello más al concepto de videofilm que al tradicional de aventura por ordenador.

La colección se inicia con un total de 8 títulos, todos ellos adaptaciones de conocidas novelas como:

«Cita con Roma» de Arthur C. Clarke.

«Farenheit 451», de Ray Bradbury.

«Amazonas» de Michel Crichton.

Perry Mason en «El Caso del Mandarín Asesino».

También hay obras tradicionales como:

«La Isla del Tesoro» o «Mago de Oz».

Los primeros títulos aparecerán a partir del mes de mayo, su formato será de diskette y el precio está por determinar.



Un millón para el mejor

En el Horal Princesa Sofía de Barcelona tuvo lugar el pasado día 11 de febrero el reparto de premios correspondiente al I Concurso SONY de Programas MSX. La numerosa asistencia puso de manifiesto el creciente interés por parte del consumidor hacia la informática, como lo demuestran los 210 programas presentados al concurso.

El premio de 1.000.000 de ptas. fue para el Instituto de Formación Profesional de El Vendrell, siendo sus autores los Sres. Jordi Francesc Rovira y Fco. Javier Rodríguez Arévalo.

SONY ESPAÑA comercializará varios de los programas enviados al concurso. Dado el espectacular éxito obtenido, el Director General de SONY ESPAÑA, Sr. Shinichi Takagi, convocó el II Concurso SONY de Programas MSX cuyas bases se publicarán próximamente.

Ratón japonés para MSX

El nuevo periférico, que aún no está en España, se denomina NEOS MS-10. Este ratón, que probablemente se comercialice con un programa de diseño gráfico, es muy sensible y muy útil en este tipo de aplicaciones, como hemos podido comprobar. Posee dos pulsadores con una función independiente cada uno. Con el de la izquierda elegimos una de las opciones del menú que aparece en pantalla, mientras que el otro nos permite volver al menú.

Las dimensiones de este ratón (34 x 103 x 63,6 mm.), facilitan en gran medida el uso por todo tipo de usuarios, ya sean estos pequeños o grandes.

Se conecta al ordenador mediante el port del joystick 1 y tiene la importante característica que se puede utilizar como joystick si el caso lo requiere.

El modo joystick se obtiene apagando el ordenador y encendiéndolo de nuevo a la vez que se mantiene pulsado el botón izquierdo del ratón. Una vez hecho esto, podremos manejar el ratón como joystick, aunque no es recomendable, ya que su función, inicialmente no es ésa. De cualquier manera, si utilizásemos el ratón en este último caso, veríamos como el objeto a desplazar lo haría a saltos (como si fuera carácter a carácter) en lugar de hacerlo suavemente (píxel a píxel).

Este lenguaje de programación, que inicialmente se considera educativo por excelencia, tiene unas posibilidades que van más allá de la educación. Pero no podemos pensar en él como un simple lenguaje para niños, puesto que sus aplicaciones abarcan aspectos muy dispares. Desde la enseñanza y complicados cálculos de ingeniería hasta el desarrollo de sistemas expertos de inteligencia artificial.

Logo:

Un lenguaje educ



Logo es un lenguaje desarrollado por el laboratorio de Inteligencia Artificial del Instituto Tecnológico de Massachusets a finales de los 60 por un equipo de investigadores dirigido por Seymor Papert. Las inquietudes de Papert no consisten tan sólo en conseguir un lenguaje eficaz en cuanto a obtener el máximo rendimiento de los ordenadores, sino, más bien, eficaz desde el punto de vista de la persona que lo usa: que pueda ser utilizado incluso por los niños, que sirva de instrumento didáctico que permita aprender y desarrollar las capacidades intelectuales de los alumnos. En definitiva, se trataba de optimizar el rendimiento educativo.

Este último hecho es el que da a *LOGO* una personalidad propia que lo diferencia del resto de los lenguajes. Papert es un matemático, pero ante todo es pedagogo, y a la hora de concebir el lenguaje estaba pensando en la educación, tratando de llevar al campo de la computación las ideas del gran psicólogo suizo Jean Piaget, creador de la escuela de Ginebra, de quien fue discípulo. Las preocupaciones iniciales de Papert se orientaban hacia la enseñanza de las matemáticas, dado el alto nivel de fracaso que se produce en esta materia y las dificultades que encontraban los niños para entenderlas. Así, pues, se trataba de desarrollar un lenguaje que permitiera a los niños «ser matemáticos» y explorar por sí mismos los conceptos abstractos con ayuda de los ordenadores.

De lo dicho hasta ahora, podría deducirse que *LOGO* es un lenguaje para uso exclusivo en la enseñanza y además sólo para niños. Nada más lejos de la realidad. Como iremos viendo a lo largo del artículo, el lenguaje *LOGO* es muy potente también desde el punto de vista informático, y el hecho de que lo puedan utilizar los niños para iniciarse en la programación o para aprender sobre otras materias es, por el contrario, un punto favorable al mismo. En el mundo de los usuarios de *LOGO*, se ha popularizado, siendo ya un lugar común, la sentencia «*LOGO* no tiene ni umbral ni techo». En efecto, la no existencia de umbral inferior lo prueban las aplicaciones desarrolladas para la educación especial y con niños muy pequeños, y el que no tenga techo queda corroborado por las complejas aplicaciones que se pueden construir en este lenguaje, desde las que requieren los más complicados cálculos matemáticos hasta sistemas expertos de Inteligencia Artificial.

Inicialmente, el lenguaje LOGO estaba orientado hacia las matemáticas, debido al alto nivel de fracaso escolar existente en esta asignatura.

El lenguaje en que se basaron los investigadores del *M.I.T.* para

desarrollar el *LOGO*, es clásico en la Inteligencia Artificial, el *LISP*, lenguaje con una sólida estructura magníficamente construida, pero de uso algo complejo para los no iniciados, especialmente por la excesiva cantidad de paréntesis que requiere.

BUSY by Santagurteña

ENTONCES GIRÉ LA TORTUGA 90 GRADOS, LUEGO LA TORTUGA ADELANTÓ 7 PIXELS, GIRÉ DE NUEVO LA TORTUGA MEDIA VUELTA Y ENTONCES LA TORTUGA SE FUE DIRECTA HAC… IMAGEN.

¿QUE SERIA DEL LOGO SIN UNA TORTUGA?

LA TORTUGA SE LLAMA CURSOR

QUE BURRO SOY

DE TODAS FORMAS ESTA SE LLAMA ENCARNA

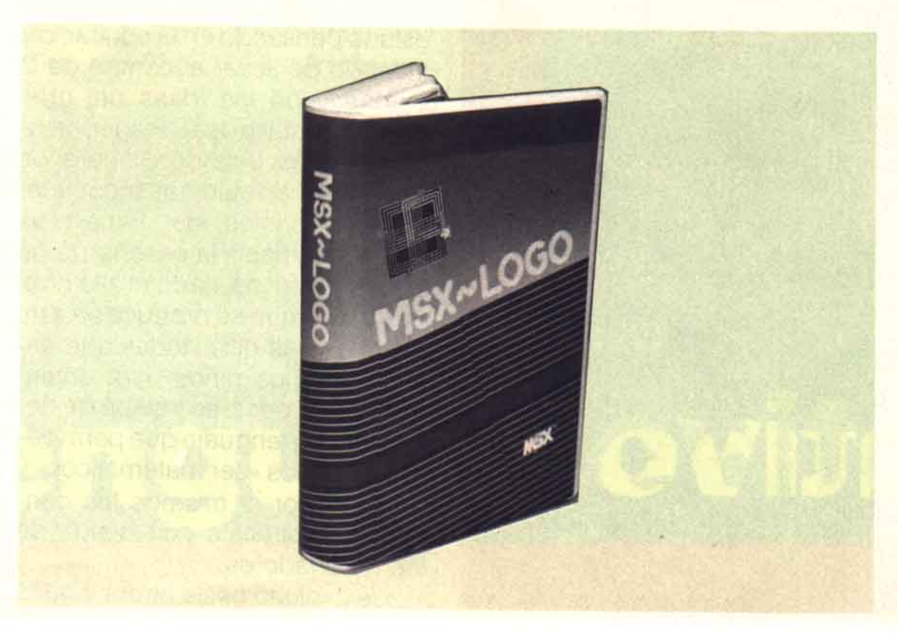
Desde sus orígenes hasta hoy se ha producido una evolución constante del interés por este lenguaje, tanto en los ámbitos educativos como de informática profesional. Las aplicaciones de *LOGO* abarcan un amplio campo de posibilidades: desde un uso alfabetizador en informática como primer lenguaje de programación hasta en cursos de Inteligencia Artificial de nivel universitario. Pero hay algo más importante, y es que su filosofía de modularización de los problemas, de estructura de los procedimientos, se impone cada día más, dando lugar a modos de trabajo diferentes con la máquina. La evolución de las sucesivas versiones de *LOGO* que aparecen cada día lo hacen más potente, alejándose mucho de aquellas primeras, muy valiosas para fines exclusivamente educativos, de Texas Instruments *(TI LOGO)* o para el *TRS-80*, dando paso a lenguajes que ya permiten manejo de ficheros, cálculo científico, y que son instrumentos capaces de satisfacer las necesidades de una gran variedad de usuarios, desde los niños hasta los investigadores de Inteligencia Artificial.

Desde sus orígenes hasta hoy, se ha producido una evolución constante en el interés por este lenguaje.

Gráficos de tortuga

Los gráficos de *LOGO* se realizan mediante la «huella»k que deja en sus desplazamientos un punto orientado denominado «tortuga». La forma de dirigir la tortuga es mediante órdenes simples, que permiten hacerla avanzar un número de pasos, retroceder, girar a la derecha un ángulo, otro a la izquierda, colocarse en una posición de coordenadas determinadas, orientarse según un rumbo dado, bajar o subir la plumilla ficticia con la que dibuja, etc. Esta idea tan simple origina muchas posibilidades gráficas y geométricas.

Los gráficos realizados por el método forman lo que se denomina en terminología *LOGO* un micromundo, de grandes posibilidades tanto geométricas como estéticas, y como instrumento para el desarrollo del pensamiento de los niños, así como una ayuda para la concreción de conceptos abstractos como los de variable, que pasarán a ser elementos útiles, ya que darán poder al alumno para realizar, por ejemplo, cuadrados de tamaño variable.



La importancia de esta idea ha llegado a trascender su ámbito original, haciendo que algunas versiones de otros lenguajes, como es el caso del *PASCAL*, las incorporen.

Interactivos

LOGO es un lenguaje que permite el desarrollo interactivo de programas, ya que las instrucciones dadas pueden ser ejecutadas directamente. Este interacción es muy importante cuando se trata de usuarios poco experimentados. Es además el hecho de poseer un intérprete lo que dio tanta popularidad al *BASIC*.

Además, *LOGO* permite una gran facilidad en la depuración de programas, al poder comprobar inmediatamente la ejecución de las distintas instrucciones. Esta facilidad de depuración de programas aumenta en algunas versiones de este lenguaje, ya que algunos poseen este tipo de comandos.

Procedimental

Una característica que comparte *LOGO* con otros lenguajes de ordenador, y en contraposición con *BASIC*, es que es procedimental. Esto significa que programar en *LOGO* consiste en enseñar procedimientos al ordenador y que, una vez los ha aprendido, el ordenador los entiende como si fueran nuevas primitivas del mismo.

Se puede decir que un procedimiento es una serie de órdenes que con un nombre dado enseñan al ordenador una nueva instrucción. Esta nueva instrucción se define mediante *PARA*, nombre seguido de la lista de órdenes que forman el cuerpo del procedimiento y acabado con un *FIN*. Si se desea introducir parámetros de entrada al procedimiento, estos han de ir a continuación del nombre precedido del signo «:». A su vez, este procedimiento prodría utilizarse para definir otro y así sucesivamente. Esta estructura procedimental permite realizar una programación modular.

Recursivo

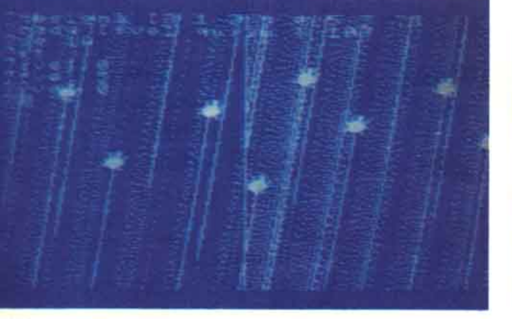
Una característica significativa del *LOGO* es su recursividad. *LOGO* favorece el uso de la recursión como técnica de programación, lo que da una gran potencia al lenguaje. Pero veamos en que consiste la recursión.

Un procedimiento se dice que es recursivo cuando se define en función de sí mismo.

Por ejemplo, el procedimiento *FIGUR* da lugar a un proceso indefinido, ya que se llama a sí mismo al final de su definición:

```
para figur: ángulo
av 25 de: ángulo
haz «ángulo»: ángulo + 2
figur: ángulo
fin
```

dando lugar a distintas figuras poligonales que se repiten indefinidamente.







El uso de la recursión al manejar listas es una de las técnicas de programación que hace de *LOGO* un lenguaje de alto poder para procesar estructuras de datos complejas, ya que un método recursivo permite expresar un problema como una versión reducida de sí mismo.

Extensible

La posibilidad de definir nuevos procedimientos, ampliando así el vocabulario que conoce el ordenador, hace que *LOGO* sea un lenguaje extensible que crece conforme a nuestras necesidades, pudiendo definir lenguajes a la medida del usuario.

Su interactividad, junto con un editor de pantalla fácil de manejar y unos mensajes de error claros hacen de *LOGO* un lenguaje de uso sencillo y agradable.

Programación modular

La estructura de *LOGO* basada, como ya hemos visto, en los procedimientos, permite realizar programas de forma modular, es decir, descomponiéndolos en partes que puedan resolverse separadamente y más simples que el problema inicial. Esta modularización puede hacerse a varios niveles, ya que cada módulo se puede, a su vez, descomponer en partes bien sencillas. Este proceso puede dar una respuesta a los planteamientos de la programación estructurada. En contraposición a otros lenguajes, que como *BASIC* dan lugar a programas poco inteligibles, *LOGO* permite y estimula la realización de programas claros, en los que cada procedimiento tiene una misión definida. En el caso de listas, de la misma forma, cada procedimiento realiza una función definida, que da una salida a otros procedimientos, permitiendo, con un estilo similar al *LISP* una técnica de programación orientada a las funciones.

LOGO para MSX

La versión de *LOGO* para ordenadores MSX desarrollada por *LCSI* (*LOGO* Computer Systems) y

distribuida en España por Philips, se presenta en un cartucho *ROM* acompañado por un completo manual de 135 páginas (comentado al final del artículo).

Quizá la característica más sorprendente de e-ta versión de *LOGO* es que se encuentra totalmente en castellano, lo cual es una gran ventaja a la hora de su aprendizaje. Nada más poner en marcha el programa, aparece un saludo y la tortuga inmóvil en el centro de la pantalla. Hay que resaltar que se trata de una auténtica tortuga, ya que las antiguas versiones de este lenguaje utilizaban un triángulo en su lugar.

Inmediatamente podemos comenzar a programar. Para ello, vamos a ver las primitivas más utilizadas del *MS-LOGO*. Estas son:

— *ALZALAPIZ, (AL):* permite alzar la plumilla para que la tortuga se desplace sin dibujar.

— *AVANZA, (AV):* hace que la tortuga se mueva en una dirección determinada y a una velocidad establecida.

— *BAJALAPIZ, (BL):* permite a la tortuga dibujar cuando está en movimiento.

— *CENTRO:* sitúa la tortuga en el centro de la pantalla.

— *FRUMBO:* fija la dirección de la tortuga.

— *FCT,* (Fija Color Tortuga): selecciona un color para nuestra amiga.

— *FCL,* (Fija Color Lápiz): selecciona un color para la tinta.

— *DERECHA, (DE):* orienta la tortuga hacia la derecha.

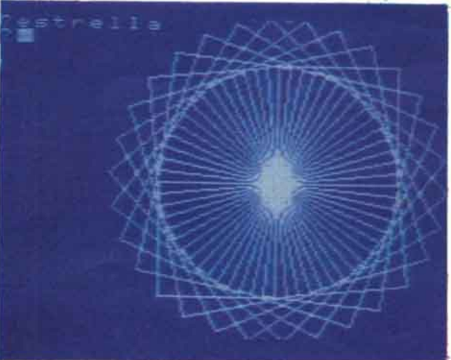
— *IZQUIERDA (IZ):* ídem, hacia la izquierda.

Sin embargo, una característica importante de este *LOGO* es que permite trabajar y programar hasta 30 tortugas diferentes, cosa que no todas las versiones lo permiten. En la pantalla podemos tener dibujando una de ellas o las 30 simultáneamente. Además, no necesariamente han de ser tortugas. El *LOGO* de Philips, tiene distintas figuras predefinidas que pueden sustituir a nuestra amiga. Podemos tener desde un perro hasta un camión, pasando por un gato, etc.

Los gráficos de LOGO se realizan mediante la «huella» que deja en sus desplazamientos una tortuga.







No obstante, la velocidad de proceso de este lenguaje no se ve mermada en modo alguno por la utilización de las 30 tortugas, es más, con una tortuga supera en velocidad la versión del PC de IBM.

Calculadora

Como todo lenguaje de programación, *LOGO* también se puede utilizar como si de una calculadora se tratase. Usando los signos o las primitivas, podremos realizar cualquier orperación aritmética. Las primitivas son las siguientes: *SUMA, DIF, PRODUCTO, COCIENTE,* que son las correspondientes a los signos +, –, * y /. La raíz cuadrada se indica con *RC* y el seno, coseno y arcotangente con; *SENO, COS* y *ARCTANG* respectivamente. También existen instrucciones que permiten obtener la parte entera de un número así como generar una secuencia de números aleatorios, que son; *REDONDEA* y *AZAR* o *REAZAR.*

Las operaciones lógicas equivalentes a *NOT, OR* y *AND* son *NO, O* e *Y.*

Tratamiento de listas

El tratamiento de listas y palabras es otro de los puntos fuertes del *LOGO,* ya que para este trabajo cuenta con 18 primitivas, aunque las más utilizadas son:

— *ASCII:* que devuelve el código *ASCII* del elemento de entrada.

— *CAR:* realiza la operación inversa de la anterior, obteniendo el carácter correspondiente al código introducido.

— *CUENTA:* produce el número de elementos del objeto de entrada.

— *FRASE:* convierte los datos de entrada en una lista única.

— *MENOSPRIMERO, (MP):* devuelve todos los elementos de entrada excepto el primero.

— *MENOSULTIMO, (MU):* ídem excepto el último.

— *VACIO:* se obtiene el valor *CIERTO (TRUE)* si en los datos de entrada tenemos una lista vacía.

Operaciones de entrada/salida

Las primitivas utilizadas en el manejo de la impresora son: *CONIMPRESORA* y *SINIMPRESORA*. En el primer caso, cualquier instrucción que se introduzca desde teclado y que aparezca en pantalla se imprimirá directamente a la impresora, la segunda instrucción vuelve todo a la normalidad. Para el uso de periféricos de almacenamiento externo (cassette o disco) existen las primitivas *GUARDA* y *CARGA* para el disco y *GUARDAC* y *CARGAC* para el cassette.

Luego existen unas primitivas especiales, que son aquellas que su uso puede provocar un *RESET* del ordenador si no se utilizan debidamente. En general, este efecto lo producen las instrucciones que escriben o leen posiciones de memoria, cuando realizamos programas en código máquina. Estas instrucciones para diferenciarlas de las primitivas normales comienzan por un punto. Alguna de estas instrucciones son:

—. *DEPOSITA:* almacena un dato en una posición determinada de memoria *(POKE* en *BASIC).*

—. *EXAMINA:* lee el dato situado en una posición concreta de la memoria *(PEEK* en *BASIC).*

—. *LLAMA:* cede el control a una subrutina en código máquina.

—. *DENTRO:* devuelve el dato leído al puerto especificado *(INP* en *BASIC).*

—. *FUERA:* envía un dato a uno de los puertos de entrada/salida *(OUT* en *BASIC).*

Es un lenguaje orientado a la enseñanza, aunque sus aplicaciones abarquen el desarrollo de sistemas expertos.



Otras posibilidades

El editor de figuras *(SPRITES)* incorporado, es otra característica importante de este programa. Este editor nos permite definir todos los *SPRITES* posibles en los ordenadores MSX, podemos incluso cambiar la forma de la tortuga. El manejo de este editor es de gran sencillez, ejecutándose mediante la instrucción *EDFI nn,* siendo *nn* el número de *SPRITE* a editar y utilizando las teclas del cursor y la barra espaciadora para efectuar cualquier cambio.



Por otro lado y dada la posibilidad de generar todo tipo de sonidos con el MSX, no podían faltar las instrucciones que lo hagan posible. Las dos primitivas que se usan para realizar esta función son; *SONIDO* y *RUIDO.*

Conclusión

En definitiva se trata de un programa útil, tanto para los que se inician en el mundo de la informática como para los usuarios más avanzados en esta materia, además de ofrecer unas amplias posibilidades didácticas, punto muy



importante para fines educativos y para aquellos padres que deseen que sus hijos hagan algo más que matar marcianos.

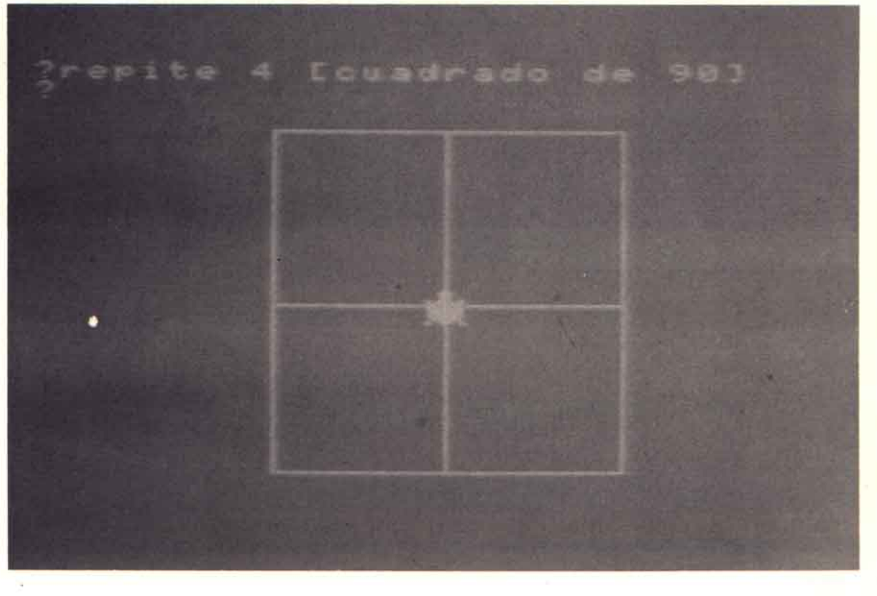
Acabaremos con un breve comentario del manual: Nos encontramos con un libro muy amplio y claramente orientado hacia los principiantes, dividido en dos partes, cada una de las cuales está dividida en varios capítulos donde se explica, con todo lujo de detalles y en un lenguaje llano, todas las características del *LOGO.*

(C) Copyright Logo Computer
Systems Inc. 1985
Bienvenido a MSX-Logo
?

La primera parte, titulada «CHARLA CON LAS TORTUGAS» consiste en una intruducción al lenguaje y enseña a dirigir las tortugas, a crear programas y a salvar en disco o cassette el trabajo realizado.

La segunda parte, titulada «GUIA RESUMIDA» contiene información sobre las características del *MS-LOGO:* colores, teclas y caracteres especiales, explicación de los procedimientos y primitivas y una lista de todas ellas organizada por grupos funcionales.

Todo esto tiene su precio, y aunque rondará las 14.000 ptas., hay que tener en cuenta que se trata de una auténtica obra de realización en todos los sentidos y necesaria para cualquiera que desee optimizar y utilizar el ordenador para algo más que jugar y programar en *BASIC.*



SOFTWARE

Programa: Dizzy Baloon
Tipo: Juego
Distribuidor: Micro Byte
Formato: Cassette

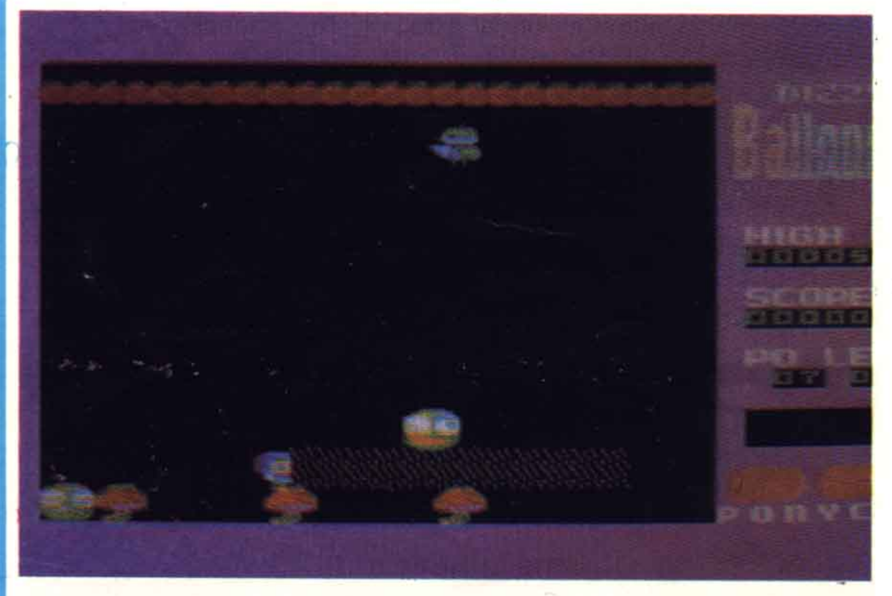
Estamos acostumbrados a ver a los animales luchando por sobrevivir en un entorno en el que la Naturaleza no les deja otra elección. Lo que sí nos puede parecer raro es que un pingüino utilice bombas de mano para distraer a sus adversarios. *Dizzy Baloon* es, sin lugar a dudas, un juego diferente.

En este juego no se trata únicamente de conseguir puntos, es necesario que a nuestro amigo, en este caso un pingüino, no le ocurra nada.

La pantalla estará dividida en tres zonas de acción. En la parte superior encontraremos una fila de manzanas que cuando comiencen a estar maduras irán cayendo para que podamos comérnoslas, pero ¡cuidado!, no todas son buenas, hay algunas con las que conseguiremos sumar puntos, y otras que nos explotarán cuando nos encuentren. La segunda zona (aunque todas son imaginarias), es en la que nos encontraremos con más peligros, ya que habrá otras frutas que de repente aparecerán y ocuparán un espacio, de tal manera que si no escapamos del entramado que formen algunas de ellas, nos veremos atrapados y se terminará el juego.

Y la tercera zona, por la que caminaremos junto con otros animalitos que si no logramos esquivar nos dejarán sin sentido y no podremos jugar. Tenemos que fijarnos bastante en la pantalla para saber en cualquier momento dónde podemos colocarnos sin que nadie o nada nos alcance, es más, no todo iba a estar en nuestra contra, podemos defendernos, no sólo con nuestra habilidad, sino que podemos lanzar bombas contra nuestros enemigos. Otro aspecto que está a nuestro favor reside en que ayudándonos de la capacidad que tenemos para saltar, podremos lanzar bombas a aquellos objetos que nos estorben, y si no conseguimos batir a nadie, al cabo de un tiempo se convertirán en frutas o setas que nos permitirán ir ascendiendo hacia las manzanas de la parte superior, aunque esto sea muy difícil conseguirlo.





Es un juego que simplemente por su realización resulta entretenido, ya que han sabido describir perfectamente en sus gráficos elementos como nuestro pingüino, las setas y multitud de frutas.

Al principio nos costará un poco familiarizarnos con el tema, pero cuando hayamos jugado algunas veces, será de los juegos que no olvidaremos.

Puntuación:
Presentación: 7
Claridad: 6
Rapidez: 7
Adicción: 6



El Universo se verá en un futuro poblado de los más extraños seres, y nosotros, los humanos, viviremos en los lugares más recónditos del espacio.

Pero el mundo futuro no sabemos como será, *Castle Combat* nos presenta una dura batalla, en la que nuestra fortaleza galáctica ha caído en manos de los *Tyrones,* despiadados guerreros del espacio que intentan de cualquier modo deshacerse de nosotros e invadir los puntos estratégicos de los humanos en el espacio para así luego, apoderarse de la Tierra.

Star Duster será la nave con la que contamos para dejar asentado el poderío de la Tierra frente a los invasores. Tendremos que tener cuidado porque no contábamos con que los *Tyrones* se apoderasen de esta fortaleza; y nuestra nave, aunque equipada con las más sútiles armas, se verá en grandes apuros ya que cada disparo tendrá que dar en un blanco, de lo contrario, mientras que ese misil se destruye, no tendremos posibilidad de volver a disparar, y los enemigos se avalanzarán hasta nosotros.

El escenario ha sido creado con gran minuciosidad, ya que los colores nos hacen situarnos en algún lugar de la superficie terrestre. El azul del cielo y la configuración del Castillo controlado por los *Tyrones,* hacen que podamos ver con más claridad cuándo se deciden a atacar y cuáles son sus posiciones, además de ser una ventaja para nosotros, pues estamos acostumbrados a este tipo de atmósfera y al medio ambiente.

Sumaremos puntos cada vez que destruyamos a uno de nuestros enemigos, pero cada vez que nos maten, volverán a formar el conjunto inicial de nueve naves enemigas, por lo que tendremos que ser muy hábiles y veloces a la hora de disparar.

Contaremos con tres naves iniciales. Si conseguimos matar a todas las naves enemigas pasaremos a una segunda pantalla aún más complicada, ya dentro del Castillo. Será ahora cuando se desarrolle la batalla, y ésta nos presentará un segundo peligro, pues los muros del Castillo nos obligarás a hacer maniobras dignas de los más audaces pilotos, así que



Programa: Castle Combat
Tipo: Juego
Distribuidor: Micro Byte
Formato: Cartucho

Castle Combat es un pionero en la informática española, en el mundo de los juegos para ordenador.

mucho cuidado.

Pero el juego no termina ahí, hay que devolver el control a la Tierra, ¿os atrevéis?

Puntuación:
Presentación: 7
Claridad: 8
Rapidez: 7
Adicción: 7

Programa: Faraón
Tipo: Juego
Distribuidor: ABC Soft
Formato: Cassette

La cultura egipcia nos ha dejado un gran legado que la humanidad ha sabido aprovechar. En diversos campos como la matemática, la física y otra multitud de aspectos, nos basamos hoy en día para crear nuestra sociedad.

Faraón, es un programa que además de acercarnos a las costumbres egipcias nos configurará como constructores y como guía de hombres. No es que vayamos a realizar una aventura, sino que nos remitiremos al Antiguo Egipto. Seremos un personaje de aquella época, nada menos que el arquitecto de las grandes pirámides. *Faraón* nos permitirá desarrollar nuestro sentido de la organización, del peligro y del poder.

Nuestro *Faraón* nos encargará que le hagamos una pirámide y nos dará medios para ello, pero todo lo demás nos lo dejará a nuestra elección. Tendremos que elegir en primer lugar la dieta que daremos a los esclavos con los que contamos en gran número, pero claro, nuestro tesoro no es infinito, habrá que administrarlo de acuerdo con las necesidades de las obras y los peligros que nos acechan, ya que en cualquier momento podemos ser atacados por las caravanas persas o alguna enfermedad puede hacer mella entre nosotros.

Si las obras son del agrado del Faraón, conseguiremos gratificaciones que pueden ser en esclavos o en dinero, y que nos sacarán de apuros.

Si por el contrario nuestra labor no le es grata, nos veremos desposeídos de todo. Estos no son todos los acontecimientos que nos pueden suceder, pero el resto te los dejaremos a tí, para que descubras la magia y la ciencia que debes emplear para salir airoso de este fin.

A medida que vayamos avanzando en el juego tendremos que decidir sabiamente ante un montón de acontecimientos, y podremos ver en el momento que queramos la situación general de las obras, de los esclavos y de nuestro tesoro.

De vez en cuando se nos presentarán pequeños conflictos, a parte de los que la propia construcción nos puede causar,y tendremos que resolver acertijos que nos permitirán seguir la construcción o no.

Pasar un rato agradable y ser un gran empresario, aunque sea de la Antigüedad, puede ser una tarea difícil pero divertida.

BIENVENIDO.
ESTAS EN EL ANTIGUO
EGIPTO BAJO EL REINADO
DE K A T M O S I S.
CUAL ES TU NOMBRE?
ABC-SOFT J. MATEOS

Puntuación:
Presentación: 7
Claridad: 8
Rapidez: 7
Adicción: 8

Programa: Idea Type
Tipo: Programa Educativo
Distribuidor: Idealogic
Formato: Cassette

Para muchos de nosotros que trabajamos, o para los estudiantes e incluso para los más pequeños en sus tareas escolares, nos es



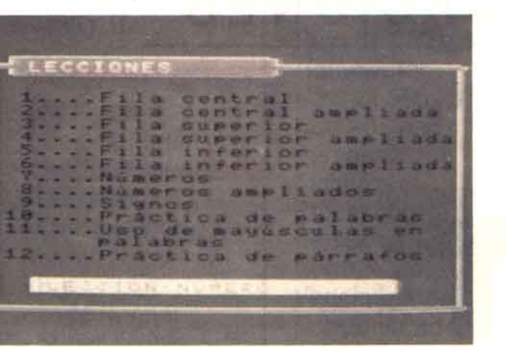
necesario escribir a máquina, pero ¿nos hemos preocupado de aprender, y aprender bien?, ahora tenemos la oportunidad, *Idea Type* es un programa educativo que tiene como fin primordial enseñar a escribir a máquina.

Los métodos que emplea el programa son totalmente visuales, ya que tanto en su apartado de orientación, donde se nos enseña a emplear una máquina de escribir, la forma de colocar los dedos, etc.; o en su apartado de ejercicios, donde realizaremos nuestros trabajos en la pantalla, nos permitirá ver con exactitud nuestros errores sin necesidad de emplear un folio tras otro.

Al haber finalizado nuestros ejercicios, que pueden ser del tiempo que nosotros queramos, recibiremos los resultados: el tiempo empleado en el ejercicio, las pulsaciones por minuto, los errores cometidos y otros datos que nos ayudarán a perfeccionar nuestro propósito.

Algunos pensarán qué diversión puede existir en escribir a máquina, aparte del placer de superarse día a día, aquí aparecen *«los cazapalabras»*, un juego que incluye el programa con el que pasaremos un rato divertido y la tarea no se nos hará tan difícil.

El que la máquina nos rete a escribir velozmente, nos hará desarrollar nuestra capacidad fotográfica y nos motivará a superarnos.



Los *«cazapalabras»*, es un divertido entretenimiento en el que cuatro muñequitos se disponen a comerse las palabras. Nosotros debemos ser más rápidos en teclearlas para que así no se las coma el *cazapalabras*, éste volverá a su punto de partida, y así nos ayudará a que consigamos más destreza y velocidad a la hora de escribir a máquina. Pasaremos de un nivel a otro sin darnos cuenta y con la práctica de este programa alguna vez por semana, conseguiremos un resultado satisfactorio.

Pero este programa educativo ha querido ayudar verdaderamente a la consecución del éxito, por lo que pasados los primeros apartados, empezaremos con párrafos y avanzaremos en el perfeccionamiento de nuestra técnica.

Es un método inigualable para aprender a escribir a máquina, y es que el mundo de la informática no tiene límites.

Puntuación:
Presentación: 7
Claridad: 9
Rapidez: 8
Adicción: 9

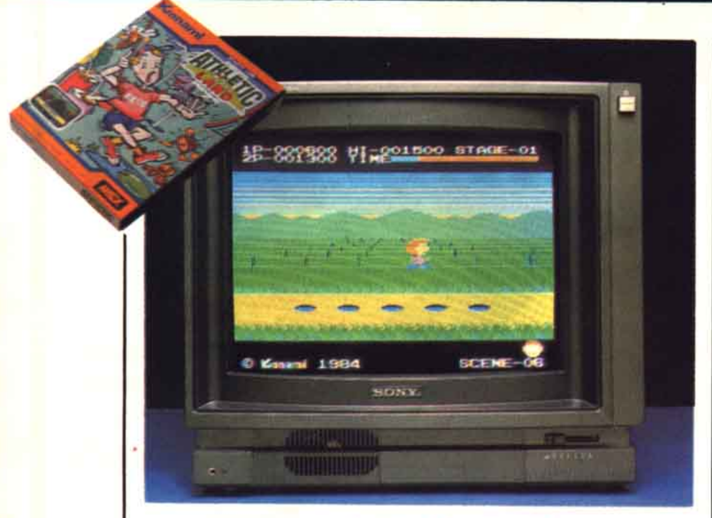
Programa: Booga Boo
Tipo: Juego
Distribuidor: Mind Games
Formato: Cassette

Es un juego dirigido a todos los vídeoadictos que sean capaces de controlar un joystick, porque de eso se trata. *Booga Boo*, es una ranita que no para de moverse y que nosotros debemos dirigirla para que alcance su meta.

Es un juego primordialmente infantil, con el que los pequeñines pasarán unos ratos divertidos y amenos en compañía de sus padres.

Los gráficos y dibujos que se han utilizado hacen que sea muy fácil el poder dirigir hacia un lado y otro a nuestra ranita, ya que contamos con la pantalla en tres dimensiones, aunque el fondo sea de color negro, pero han sabido diferenciar, la situación de los elementos.

Programas Sony MSX, para lo que guste ordenar.



JUEGOS



GESTION



EDUCATIVOS



APLICACIONES





Al comenzar el juego, nuestro animalito caerá, sin poder hacer nada por impedirlo, al fondo del suelo cubierto por un entramado de flores y plantas que no le dejarán ver siquiera la luz.

Debemos hacer saltar a la ranita para que consiga subir por la multitud de ramas y árboles que la permitirán llegar a su lugar de origen y no volver a recordar esa horrible pesadilla. Parece un juego fácil a primera vista, pero no es así. Cada salto que consigamos que dé la ranita, tendremos que dirigirlo lo más acertadamente posible, porque en el fondo del precipicio en el que ha caído existen plantas carnívoras que se la comerían y se acabaría la partida.

Otro problema será cuando aparezca el peligro más grande de todos. En ese precipicio parece que no ha pasado el tiempo y existen los más raros ejemplares de la época antediluviana, es el caso de un pájaro que nos acechará constantemente y que si no nos resguardamos bien caeremos presas de sus garras y nos llevará a que nos devoren las plantas carnívoras del fondo del pantano.

La forma de puntuar en el juego dependerá del nivel que alcancemos subiendo, antes de que alguno de los peligros nos lo impida. Si hemos conseguido ascender bastantes metros hacia la luz, la puntuación será muy alta, sino, tendremos que volver a jugar para superarnos.

Booga Boo es un juego informal y divertido, interesante para cualquier edad y un buen pasatiempo para los pequeñines.

Puntuación:
Presentación: 8
Claridad: 8
Rapidez: 8
Adicción: 9



PROGRAMAS SONY MSX

Educativos

- Monkey Academy
- Alfamat
- Viaje Espacial
- Multipuzzle
- Noria de Números
- Corro de Formas
- Coconuts
- Yo Calculo
- Selva de Letras
- El Cubo Informático
- Electro-graf
- El Rancho
- Teclas Divertidas
- Boing Boning
- Compulandia
- Mil Caras
- Logo
- Países Mundo-1
- Países Mundo-2
- Tutor
- Computador Adivino
- Aprend. Inglés-1
- Aprend. Inglés-2
- Cosmos
- Curso de Básic
- Juego de Números

Juegos

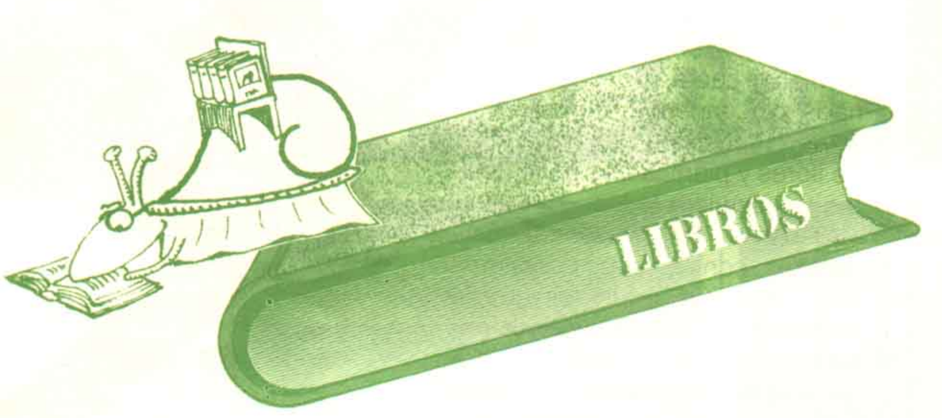
- Antártic Adventure
- Athletic Land
- Sparkie
- Juno First
- Car Jamboree
- Battle Cross
- Crazy Train
- Mouser
- Computer Billiards
- Alí Babá
- Track & Field-I
- Track & Field-II
- Dorodon
- Chess (Ajedrez)
- Senjo
- E.I.
- Lode Runner
- Super Tennis
- Backgammon
- Super Golf
- Hustler
- Binary Land
- Driller Tanks
- Stop the Express
- Ninja
- Les Flics
- La Pulga
- The Snowman
- Cubit
- Pack 16K
- Fútbol
- Kung Fu
- Batalla Tanques
- Mr. Wong
- Xixolog
- Buggy
- Sweet Acorn
- Peetan
- Jump Coaster
- Buggy 84
- 3D Water Driver
- Pinky Chase
- Wedding Bells
- Fightting Rider

Aplicación

- Memoria Ram 4 K
- Creative Greetings
- Character Collect
- Quinielas y Reducciones
- Pascal
- Ensamblador
- Generador Juegos

Gestión

- Hoja de Cálculo
- Homewriter
- Control Stocks
- Contabilidad Personal
- Ficheros
- Procesador de Textos
- Control Stocks
- Vencimientos
- Contabilidad 1.500



Título: Guía del programador MSX
Autor: C. I. Burkinshaw
Editorial: RA-MA
Páginas: 208

Cualquier libro, que ayude para introducirnos en el lenguaje del sistema MSX, es necesario consultarlo. Pero si además de sernos útil en ese tema, nos proporciona ayuda en todo lo referente al ordenador y el mundo que le rodea, adquiere un matiz de necesario.

Guía del programador MSX, es un libro creado fundamentalmente para aquellos que quieran conocer en profundidad las posibilidades de su ordenador y poder solucionar todos los problemas que se nos puedan plantear a la hora de hacer un programa o aplicar todas las características y elementos que componen este sistema informático.

Un elemento primordial a la hora de trabajar con un ordenador es conocer el lenguaje en el que trabaja éste, como es el caso del BASIC.

¿Pero qué hay detrás del BASIC y de dónde procede? Verdaderamente, muchos de nosotros hemos comenzado a programar sin ni siquiera saber qué son los números binarios, las variables numéricas o las alfanuméricas.

Este libro se ha encargado de realizar un apartado en el que se nos habla con detenimiento de las raíces del BASIC y de los sistemas operativos que utiliza el ordenador, para que la figura del



programador no se limite al mero acto de crear programas, sino que sea un verdadero conocedor de la ciencia en la que y con la que está trabajando, y por supuesto que pueda conocer todos los problemas y posibilidades de un ordenador.

Apartados dedicados al Z-80, al vocabulario y a la configuración del sistema MSX, hacen de él un libro técnico e informativo que describe al ordenador en un marco de ayuda al programador, enseñándole y «educándole» a ver las cosas desde una perspectiva informática.

La confección del libro no ha escatimado en incluir dibujos que nos puedan ayudar a ver las cosas desde un punto de vista mucho más claro, sin tener que leernos amplios manuales que traten de elementos en sí.

Formación de gráficos, sprites y colores se encuentran resumidos en el precesador de vídeo (VDP), una definición amplia y concisa referente a todos accesos y modos de procesarlos.

No hemos ennumerado todos los apartados de los que consta el libro, pero hemos descrito por encima los más importantes.

Guía del programador MSX, es necesario para conocer a fondo el ordenador y sus elementos dentro del sistema MSX.

Título: Microordenadores y Cassettes
Autor: Mike Salem
Editorial: Noray
Páginas: 75

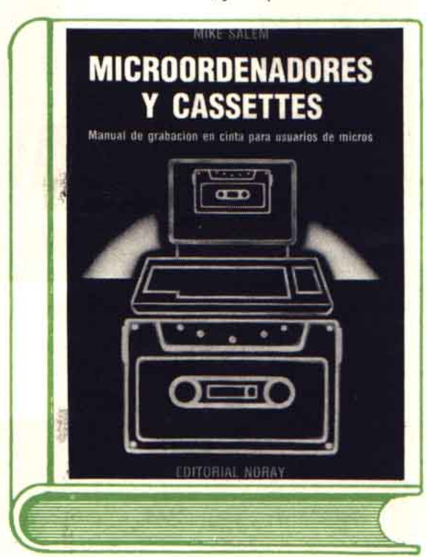
Esta obra trata de ayudar a los usuarios de ordenadores, en este caso microordenadores, que utilizan un sistema de cinta como almacenamiento externo de programas y datos. Va dirigido a los usuarios de ordenador, que empiezan ahora a familiarizarse con los microordenadores u ordenadores domésticos, en cuyo caso el cassette es el periférico común en todos los lugares.

«Microordenadores y Cassettes», no se limita al simple aspecto de describirnos las utiidades y aplicaciones del cassette, sino que intenta dar una visión de conjunto de los demás periféricos, que tienen como misión re-

producir, grabar o guardar programas, como es el caso de los cartuchos, microdrives o cintas en bobinas.

Pero centrándonos en el tema base del libro, las cintas de cassette, nos encontramos capítulos como: la elección del grabador, comprobación y ajuste, selección y cuidado de cintas, grabaciones fiables, carga de cintas difíciles, accesorios, cómo trabajan los grabadores etc., lo que nos da una visión clara del contenido de esta obra, que nos permitirá terminar con momentos desagradables como que se nos borre un programa o que no nos cargue una cinta.

El libro contiene explicaciones referidas al tipo de cassette que debemos utilizar. El tema de los cabezales viene explicado de forma detallada, ya que todos sa-



bemos que la mala utilización de una cinta puede hacer que el programa se borre. Este es el caso de la utilización de cabezas estéreo que tienen la mayoría de los aparatos.

Los gráficos y el lenguaje tan descriptivo que se utiliza en el texto hacen que sea un libro interesante y además beneficioso para completar nuestro aprendizaje en el manejo de ordenadores periféricos.

Título: MSX Programas y Utilidades
Autor: Lüers
Editorial: Ferre Moret, S. A.
Páginas: 196

Este libro está dedicado a todas las personas que dedican su trabajo al estudio de la informática, sus aplicaciones y utilidades.

Los circuitos interiores del ordenador, aún hoy en día, son para muchos totalmente desconocidos por ser éste un aspecto que se ha dejado un tanto de lado. Este libro trata de introducirnos en todos los pormenores del ordenador y nos ayuda a perfeccionar nuestros trabajos y nuestra formación.

Apartados referidos a las memorias y los editores, nos facilitan la labor a la hora de saber las posibilidades de nuestro ordenador, pero en este tipo de temas no se puede hacer una mera descripción de los elementos, sino que, como muy bien han sabido plasmar en este libro, se enumeran uno por uno y se da una explicación detallada de ellos.

Puntualizar sobre las direcciones de memoria o las abreviaturas de las palabras y tokens, son hechos que de forma sistemática debemos conocer, pero todo esto nos lleva de una manera inductiva a la materialización del hecho que, en una palabra, se denomina aplicación o programa.

Vamos a conocer un poco las partes de las que consta un programa, referido al caso de los juegos de ordenador:

— Una masa de DATAS casi interminable para el juego de caracteres.

— La rutina de proceso y de carga para los DATAS.

Su ordenador MSX puede representar por pantalla un máximo de 256 caracteres simultáneamente. Por tanto, con este programa se pueda proceder del siguiente modo: convertir el jue-



go de caracteres completo en la escritura del ordenador o trabajar con escritura normal y escritura del ordenador al mismo tiempo. Es así que intercambiando pokes del RAM de vídeo conseguimos crear un juego de caracteres completamente nuevo.

Pero esto es una pequeña muestra de las utilidades que nos proporciona tener este libro, ya que en su mayor extensión nos habla de los programas que podemos crear. La forma más correcta de hacerlo es conocer un poco ese conglomerado de circuitos y memorias que a veces nos dejan atónitos.

La forma en la que está escrito y los métodos empleados nos llevan a conocer en este libro las aplicaciones más inmediatas y las mejores utilidades a la hora de programar y conocer nuestras posibilidades.



Seikosha SP-1000MX



Entre las muchas impresoras existentes para ordenadores, encontramos algunas especialmente adaptadas para cumplir los requisitos del estándar MSX, ese es el caso de la Seikosha SP-1000MX.

Se trata de una impresora matricial con unas prestaciones suficientemente buenas como para satisfacer las necesidades de todos los usuarios.

La impresión es de muy buena calidad y permite utilizar las más diversas opciones como veremos más adelante.

Exteriormente, da sensación de solidez y fiabilidad. Tanto las teclas de control como los diversos interruptores y *switches* se encuentran a mano, esto último es de agradecer, puesto que en muchos casos, hay que abrir la impresora para llegar a los *switches* y cambiarlos. Estos están situados en la parte posterior del aparato. También queda a la vista el mecanismo que selecciona la posibilidad de utilizar tracción o arrastre del papel. Esta última posibilidad es utilísima, sobre todo para el usuario que utiliza su ordenador para aplicaciones o fines comerciales. Es de destacar, que cuando se utiliza la opción de tracción, el papel se inserta verticalmente y el ordenador lo introduce directamente y lo prepara para empezar a escribir.

El juego de caracteres es el estándar MSX, incluyendo todos los símbolos especiales como la *ñ* y todos los caracteres gráficos.

El bajo nivel de ruido es una característica que resulta agradable, sobre todo cuando se utiliza la im-



Figura 1: Una impresora polivalente, tanto para informes como para listados de programas.



Figura 2: Los interruptores, fácilmente accesibles, permiten la obtención de cualquier tipo de letra.

presora durante prolongados períodos de tiempo.

Dispone de un amplio repertorio de posibilidades de impresión.

En el modo haxadecimal, no se imprimen los caracteres que son envíados a la impresora, sino que imprime sus correspondientes códigos *ASCII* en notación haxadecimal. En doble ancho, permite la realización de caracteres ampliados, útil para realizar los títulos de programas o documentos.

Con la letra comprimida, se escriben los caracteres reducidos, facilitando la impresión de largos listados con el consiguiente ahorro de papel.

La letra de alta calidad, proporciona una letra similar a la de una máquina de escribir.

También tenemos la posibilidad de utilizar letra en negrita, subrayado, cursiva, etc., además de poder combinar todos estos casos para obtener resultados únicos.

El manual que acompaña la impresora es muy completo, con una pequeña laguna: está escrito en inglés. Sin embargo, salvado este obstáculo, se muestran todas las características del aparato, así como sus posibilidades, incluyendo pequeños programas de ejemplo.

FICHA TECNICA

Dimensiones: 390 x 119 x 266 mm.
Consumo de potencia: 30W. (durante auto test).
Método de impresión: impacto de puntos.
Cabeza de impresión: de 9 puntos.
Juego de caracteres: estándar MSX.
Arrastre de papel: tracción y fricción.
Precio: 64.900 ptas + IVA.



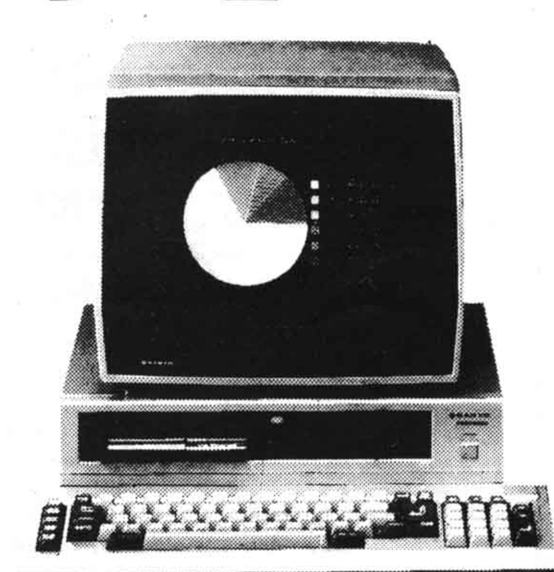
Es, por definición, laboriosa, económica y buena administradora. Y si además de hormiga es japonesa, su capacidad de trabajo sobrepasa los límites de lo imaginable.

Nuestros Ordenadores SANYO son así. Trabajan como los mejores ordenadores y son eficientes como los mejores japoneses.

Ordenadores, pequeños, silenciosos, fieles, infatigables. Y muy personales. Tan personales como para-usted, por ejemplo.

El Ordenador SANYO Serie 550.

La hormiga japonesa



- **CPU 8088 de 16 bits. Sistema Operativo MS-DOS.**
- **Disponible con uno (MBC-550-2) o dos (MBC-555-2) diskettes de 5 1/4" para mayor versatilidad en proceso y almacenaje de datos.**
- **Pantallas de alta resolución con gráficos de 640 × 200 puntos monocroma (CRT-36) o color (CRT-70). Asignación de uno de los ocho colores a cada punto.**
- **Memoria RAM 128KB, expandible a 256KB para trabajos de alta velocidad de proceso y grandes cantidades de información. También permite el uso de sofisticados lenguajes de programación.**
- **Compacto. Diseño funcional con teclado separado que permite al usuario operar en cualquier posición de trabajo.**

Deseo recibir más información sobre los ordenadores SANYO

Modelo ____________________

Nombre ____________________

____

Empresa ____________________

Cargo ____________________

Dirección ____________________

Nº ____________________

Ciudad ____________________

Provincia ____________________

Envíe este cupón a SANYO INFORMATICA, S.A.
Pº Valldaura, 256-258. 08016 Barcelona

SANYO
ORDENADORES

Problemas de electricidad



Los problemas de electricidad son una constante en la vida cotidiana. Algunos «manitas» se dedican a los menesteres de montajes de aparatos, equipos, etc., con lo que todo lo que hayan aprendido del tema lo sabrán por experiencia. Sin embargo, los no iniciados en la materia, verán en los temas de electricidad algo de brujería, lo que explica la dificultad de su aprendizaje.

Este programa, con sus cinco opciones, expone diversos problemas de electricidad que nos podemos encontrar en cualquier situación. Cada uno de ellos muestra la fórmula y la definición correspondiente, así como las otras opciones que van de acuerdo con el problema en cuestión.

En el programa, hay que destacar la generación y utilización de letras en vídeo inverso. Para ello desde las líneas 120 hasta la línea 1.270, se han definido, con la instrucción VPOKE las letras de la A hasta la U en vídeo inverso (esto es, el fondo blanco y la letra en negro). Este programa funcionará en ordenadores MSX con una memoria de más de 32K.

Diego Castaño Rodríguez
Sevilla

```
10 '----------------------------
20 '-PROBLEMAS DE ELECTRICIDAD-
30 '- DIEGO CASTAÑO RODRIGUEZ -
40 '-      PARA MSX-MAGAZINE    -
50 '----------------------------
60 '
70 '--DEFINICION DE CARACTERES--
80 '--      DE VIDEO INVERSO    --
90 '
100 SCREEN1:COLOR5,1,1:WIDTH32:KEY OFF
110 '--DEFINIR LETRA A--
120 VPOKE 776,&B11111111
130 VPOKE 777,&B11001111
140 VPOKE 778,&B10110111
150 VPOKE 779,&B10110111
```

```
160 VPOKE 780,&B10000111
170 VPOKE 781,&B10110111
180 VPOKE 782,&B10110111
190 VPOKE 783,&B11111111
200 '--DEFINIR LETRA D--
210 VPOKE 800,&B11111111
220 VPOKE 801,&B10001111
230 VPOKE 802,&B10110111
240 VPOKE 803,&B10110111
250 VPOKE 804,&B10110111
260 VPOKE 805,&B10110111
270 VPOKE 806,&B10001111
280 VPOKE 807,&B11111111
290 '--DEFINIR LETRA E--
300 VPOKE 808,&B11111111
310 VPOKE 809,&B10000111
320 VPOKE 810,&B10111111
330 VPOKE 811,&B10001111
340 VPOKE 812,&B10111111
350 VPOKE 813,&B10111111
360 VPOKE 814,&B10000111
370 VPOKE 815,&B11111111
380 '--DEFINIR LETRA G--
390 VPOKE 824,&B11111111
400 VPOKE 825,&B11001111
410 VPOKE 826,&B10110111
420 VPOKE 827,&B10111111
430 VPOKE 828,&B10100111
440 VPOKE 829,&B10110111
450 VPOKE 830,&B11001111
460 VPOKE 831,&B11111111
470 '--DEFINIR LETRA I--
480 VPOKE 840,&B11111111
490 VPOKE 841,&B11011111
500 VPOKE 842,&B11011111
510 VPOKE 843,&B11011111
520 VPOKE 844,&B11011111
530 VPOKE 845,&B11011111
540 VPOKE 846,&B11011111
550 VPOKE 847,&B11111111
560 '--DEFINIR LETRA J--
570 VPOKE 848,&B11111111
580 VPOKE 849,&B11100011
590 VPOKE 850,&B11110111
600 VPOKE 851,&B11110111
610 VPOKE 852,&B10110111
620 VPOKE 853,&B10110111
630 VPOKE 854,&B11001111
640 VPOKE 855,&B11111111
650 '--DEFINIR LETRA L--
660 VPOKE 864,&B11111111
670 VPOKE 865,&B10111111
680 VPOKE 866,&B10111111
690 VPOKE 867,&B10111111
700 VPOKE 868,&B10111111
710 VPOKE 869,&B10111111
720 VPOKE 870,&B10000111
730 VPOKE 871,&B11111111
740 '--DEFINIR LETRA M--
750 VPOKE 872,&B11111111
760 VPOKE 873,&B10111011
770 VPOKE 874,&B10010011
780 VPOKE 875,&B10101011
```

```
790 VPOKE 876,&B10111011
800 VPOKE 877,&B10111011
810 VPOKE 878,&B10111011
820 VPOKE 879,&B11111111
830 '---DEFINIR LETRA N--
840 VPOKE 880,&B11111111
850 VPOKE 881,&B10011011
860 VPOKE 882,&B10011011
870 VPOKE 883,&B10101011
880 VPOKE 884,&B10101011
890 VPOKE 885,&B10110011
900 VPOKE 886,&B10110011
910 VPOKE 887,&B11111111
920 '---DEFINIR LETRA O--
930 VPOKE 888,&B11111111
940 VPOKE 889,&B11001111
950 VPOKE 890,&B10110111
960 VPOKE 891,&B10110111
970 VPOKE 892,&B10110111
980 VPOKE 893,&B10110111
990 VPOKE 894,&B11001111
1000 VPOKE 895,&B11111111
1010 '---DEFINIR LETRA P--
1020 VPOKE 896,&B11111111
1030 VPOKE 897,&B10001111
1040 VPOKE 898,&B10110111
1050 VPOKE 899,&B10001111
1060 VPOKE 900,&B10111111
1070 VPOKE 901,&B10111111
1080 VPOKE 902,&B10111111
1090 VPOKE 903,&B11111111
1100 '---DEFINIR LETRA R--
1110 VPOKE 912,&B11111111
1120 VPOKE 913,&B10001111
1130 VPOKE 914,&B10110111
1140 VPOKE 915,&B10001111
1150 VPOKE 916,&B10101111
1160 VPOKE 917,&B10110111
1170 VPOKE 918,&B10110111
1180 VPOKE 919,&B11111111
1190 '---DEFINIR LETRA U--
1200 VPOKE 936,&B11111111
1210 VPOKE 937,&B10110111
1220 VPOKE 938,&B10110111
1230 VPOKE 939,&B10110111
1240 VPOKE 940,&B10110111
1250 VPOKE 941,&B10110111
1260 VPOKE 942,&B11001111
1270 VPOKE 943,&B11111111
1280 '---CABECERA DE PRESENTACION--
1290 FORX=0TO23STEP4
1300 FORY=0TO32:LOCATEY,X:PRINT"‾":NEXTY,X
1310 FORW=2TO14STEP4:FORZ=1TO32:LOCATEZ,W:PRINT"-":NEXTZ,W
1320 FORX=0TO19:LOCATE 31,X:PRINT"■":NEXT
1330 FORX=0TO19:LOCATE 0,X:PRINT"■":NEXT
1340 LOCATE 2,1:PRINT"ordenador:"
1350 LOCATE 3,3:PRINT"COMPATIBLE MSX."
1360 LOCATE 2,5:PRINT"memoria:"
1370 LOCATE 3,7:PRINT"32 Kb."
1380 LOCATE 2,9:PRINT"programa:"
1390 LOCATE 3,11:PRINT"PROBLEMAS DE ELECTRICIDAD."
1400 LOCATE 2,13:PRINT"lenguaje:"
1410 LOCATE 3,15:PRINT"BASIC MSX."
```

```
1420 LOCATE 7,17:PRINT"┌──────────────┐"
1430 LOCATE 7,18:PRINT"|(c) D.C.R. 1,985 |"
1440 LOCATE 7,19:PRINT"└──────────────┘"
1450 FORDE=0TO2000:NEXTDE
1460 LOCATE 3,21:PRINT"PULSA UNA TECLA PARA SEGUIR."
1470 IF INKEY$=""THEN1470
1480 '
1490 '---menú principal---
1500 '
1510 SCREEN1:WIDTH32:CLS:COLOR 5,1,1:KEY OFF
1520 PRINT"┌───────────────────────┐"
1530 PRINT"|  PROBLEMAS DE ELECTRICIDAD  |"
1540 PRINT"├───────────────────────┤"
1550 PRINT"|  (c) DIEGO CASTAÑO. 1.985   |"
1560 PRINT"├───────────────────────┤"
1570 PRINT"|                             |"
1580 PRINT"| -1 Fórmula de la resistencia|"
1590 PRINT"|    en función de la longitud|"
1600 PRINT"|    y la sección.            |"
1605 COLOR 5
1610 PRINT"|                             |"
1620 PRINT"| -2 Ley de Ohm.              |"
1630 PRINT"|                             |"
1640 PRINT"| -3 Potencia de disipación.  |"
1650 PRINT"|                             |"
1660 PRINT"| -4 Resistencias en paralelo.|"
1670 PRINT"|                             |"
1680 PRINT"| -5 Resistencias en serie.   |"
1690 PRINT"|                             |"
1700 PRINT"|                             |"
1710 PRINT"|  Numero de opción...        |"
1720 PRINT"|            ┌───────┐        |"
1730 PRINT"└────────────┤ M S X ├────────┘"
1740 PRINT"             └───────┘"
1750 A$=INKEY$:IFA$=""THEN1750
1760 BEEP
1770 IFA$<"1" OR A$>"5"THEN1750
1780 ON VAL(A$) GOTO 1820,2630,3260,3960,4780
1790 '
1800 '---comienza la opción nº1
1810 '
1820 SCREEN0:WIDTH40:COLOR5,1,1:KEY OFF
1830 PRINT"FORMULA DE LA RESISTENCIA DE UN CONDUC-"
1840 PRINT"TOR EN FUNCION DE SU LONGITUD Y DE SU "
1850 PRINT"SECCION"
1860 PRINT"-------------------------------------- "
1870 PRINT"La resistencia de un hilo conductor de-"
1880 PRINT"pende de su longitud, de su sección y"
1890 PRINT"de la naturaleza del conductor.Se mide"
1900 PRINT"en ohmios(Ω).   "
1910 PRINT"La fórmula de la resistencia en función"
1920 PRINT"de la longitud y sección del conductor "
1930 PRINT"es:"
1940 PRINT"          ┌ L=Longitud en metros."
1950 PRINT"     L    |"
1960 PRINT"R=p ---   ├ S=Sección en m/m².'"
1970 PRINT"     S    |"
1980 PRINT"          └ p=Resistividad.         ":PRINT
1990 PRINT"             ┌───────────────┐"
2000 PRINT"             | RESISTIVIDAD |  "
2010 PRINT"┌────────────┴───────────────┴──┐"
2020 PRINT"| Cobre= 0.017; Aluminio= 0.028 |"
2030 PRINT"└───────────────────────────────┘"
```

```
2040 LOCATE 0,22:PRINT"Pulsa una tecla para continuar."
2050 IF INKEY$=""THEN2050
2060 SCREEN0:CLS
2070 PRINT"┌──────────────────────────────────────┐"
2080 PRINT"|            FORMULAS  BASICAS         |"
2090 PRINT"├──────────────────────────────────────┤"
2100 PRINT"|                L                     |"
2110 PRINT"| 1- R=p ───                           |"
2120 PRINT"|                S                     |"
2130 PRINT"|                                      |"
2140 PRINT"|                S                     |"
2150 PRINT"| 2- L=R ───                           |"
2160 PRINT"|                p                     |"
2170 PRINT"|                                      |"
2180 PRINT"|                L                     |"
2190 PRINT"| 3- S=p ───                           |"
2200 PRINT"|                R                     |"
2210 PRINT"|                                      |"
2220 PRINT"| 4- RETORNO AL MENU.                  |"
2230 PRINT"|                                      |"
2240 PRINT"| Numero de opción...                  |"
2250 PRINT"└──────────────────────────────────────┘"
2260 B$=INKEY$:IFB$=""THEN2260
2270 BEEP
2280 IFB$<"1"ORB$>"4"THEN2260
2290 ON VAL(B$) GOTO2340,2430,2520,1480
2300 LOCATE 0,22:PRINT"Pulsa una tecla para volver al menú."
2310 IF INKEY$=""THEN2310
2320 BEEP
2330 GOTO2060
2340 SCREEN0:WIDTH40:CLS
2350 LOCATE1,1:PRINT"     L"
2360 LOCATE1,2:PRINT"R=p ─  "
2370 LOCATE1,3:PRINT"     S"
2380 PRINT:INPUT"LONGITUD EN METROS: ";L
2390 PRINT:INPUT"SECCION EN m/m² : ";S
2400 PRINT:INPUT"RESISTIVIDAD (p): ";P
2410 PRINT:PRINT"RESISTENCIA=";CSNG(P*(L/S));" OHMIOS"
2420 GOTO2300
2430 CLS
2440 LOCATE1,1:PRINT"     S"
2450 LOCATE1,2:PRINT"L=R───"
2460 LOCATE1,3:PRINT"     p "
2470 PRINT:INPUT"SECCION EN m/m² : ";S
2480 PRINT:INPUT"RESISTIVIDAD (p): ";P
2490 PRINT:INPUT"RESISTENCIA EN Ω: ";R
2500 PRINT:PRINT"LONGITUD=";CSNG(R*(S/P));" METROS"
2510 GOTO 2300
2520 CLS:LOCATE1,1:PRINT"     L"
2530 LOCATE1,2:PRINT"S=p ──"
2540 LOCATE1,3:PRINT"     R "
2550 PRINT:INPUT"LONGITUD EN METROS: ";L
2560 PRINT:INPUT"RESISTENCIA EN Ω: ";R
2570 PRINT:INPUT"RESISTIVIDAD (p): ";P
2580 PRINT:PRINT"SECCION=";CSNG(P*(L/R));" m/m² ."
2590 GOTO 2300
2600 '
2610 '---comienza la opción nº 2---'
2620 '
2630 SCREEN2:COLOR 5,1,1:CLS
2640 CLOSE:OPEN"GRP:"AS#1
2650 DRAW"BM80,10":PRINT#1,"──LEY DE OHM──"
2660 DRAW"BM10,20":PRINT#1,"La diferencia de potencial apli cada a los extremos
```

```
de un condu ctor,expresada en VOLTIOS,es i- gual al producto de la resisten cia
de dicho conductor, dada en OHMIOS, por la intensidad que   por él circula, expr
esada en AM "
2670 DRAW"BM7,77":PRINT#1,"PERIOS."
2680 DRAW"BM10,130C5R20U5R40D5NR20D5L40U5BM21,125C5E6R17E6F6R17F6":PAINT(50,130)
,5
2690 DRAW"BM48,102":PRINT#1,"V"
2700 DRAW"BM10,140":PRINT#1,"—➧":DRAW"BM15,145":PRINT#1,"I":DRAW"BM48,127":COLO
R 1:PRINT#1,"R"
2710 DRAW"BM100,100":COLOR2:PRINT#1,"A-  V=I*R (VOLTIOS)"
2720 DRAW"BM100,130":PRINT#1,"B-  I=V/R (AMPERIOS)"
2730 DRAW"BM100,160":PRINT#1,"C-  R=V/I  (OHMIOS)
2740 DRAW"BM0,185":COLOR 5:PRINT#1,"Pulsa una tecla para continuar."
2750 IF INKEY$=""THEN2750
2760 SCREEN0:WIDTH40:COLOR 5,1,1:CLS
2770 PRINT"┌──────────────────────────────────┐"
2780 PRINT"|           ---LEY DE OHM---        |"
2790 PRINT"├──────────────────────────────────┤"
2800 PRINT"|                                  |"
2810 PRINT"|  1- RESISTENCIA EN OHMIOS        |"
2820 PRINT"|                                  |"
2830 PRINT"|  2- TENSION EN VOLTIOS           |"
2840 PRINT"|                                  |"
2850 PRINT"|  3- INTENSIDAD EN AMPERIOS       |"
2860 PRINT"|                                  |"
2870 PRINT"|  4- RETORNO AL MENU              |"
2880 PRINT"|                                  |"
2890 PRINT"|                                  |"
2900 PRINT"|  Número de opción...             |"
2910 PRINT"└──────────────────────────────────┘"
2920 A$=INKEY$
2930 IF A$=""THEN 2920
2940 BEEP
2950 IF A$<"1" OR A$>"5" THEN 2920
2960 ON VAL(A$) GOTO 3020,3090,3160,1480
2970 LOCATE 0,19
2980 PRINT"Pulsa una tecla para volver al menú."
2990 IF INKEY$="" THEN 2990
3000 BEEP
3010 GOTO 2760
3020 CLS
3030 PRINT" R(Ω)=V/I"
3040 PRINT:PRINT
3050 INPUT" TENSION ";V:PRINT
3060 INPUT" INTENSIDAD";I:PRINT
3070 PRINT"RESISTENCIA=";CSNG(V/I);"OHMIOS"
3080 GOTO 2970
3090 CLS
3100 PRINT" V(v)=R*I
3110 PRINT:PRINT
3120 INPUT" RESISTENCIA ";R:PRINT
3130 INPUT" INTENSIDAD ";I:PRINT
3140 PRINT"TENSION=";CSNG( R*I);"VOLTIOS"
3150 GOTO 2970
3160 CLS
3170 PRINT" I(A)=V/R"
3180 PRINT:PRINT
3190 INPUT" TENSION ";V:PRINT
3200 INPUT" RESISTENCIA ";R:PRINT
3210 PRINT"INTENSIDAD=";CSNG(V/R);"AMPERIOS"
3220 GOTO 2970
3230 '
3240 '---comienza la opción nº 4---
```

```
3250 '
3260 SCREEN0:COLOR5,1,1:CLS:WIDTH40
3270 PRINT "     ---- POTENCIA DE DISIPACION ----":PRINT
3280 PRINT" Cuando a través de una resistencia cir-cula una corriente, la energi
a electrica se disipa en forma de calor y la poten-cia disipada se calcula por l
a ecuación:"
3290 PRINT:PRINT"                     P=V*I"
3300 PRINT:PRINT" siendo I la intensidad de la corriente en AMPERIOS y V la d.d.
p. aplicada a la resistencia, en VOLTIOS."
3310 PRINT" Dado que V e I están relacionados con  el valor de la resistencia R(
ley de Ohm), tambien puede calcularse la potencia  por medio de éste par de fórm
ulas:"
3320 PRINT:PRINT"    P=R*I²         P=V²/R"
3330 LOCATE 0,22:PRINT"Pulsa una tecla para continuar."
3340 IF INKEY$=""THEN3340
3350 CLS:SCREEN0:WIDTH40
3360 PRINT"┌─────────────────────────────────────┐"
3370 PRINT"|     ---- POTENCIA DE DISIPACION----  |"
3380 PRINT"├─────────────────────────────────────┤"
3390 PRINT"|                                      |"
3400 PRINT"| 1 - P(W)=V*I                         |
3410 PRINT"|                                      |"
3420 PRINT"| 2 - P(W)=R*I²                        |"
3430 PRINT"|                                      |"
3440 PRINT"| 3 - P(W)=V²/R                        |"
3450 PRINT"|                                      |"
3460 PRINT"| 4 - Imax=SQR(P/R)                    |"
3470 PRINT"|                                      |"
3480 PRINT"| 5 - Vmax=SQR(P*R)                    |"
3490 PRINT"|                                      |"
3500 PRINT"| 6 - RETORNO AL MENU                  |"
3510 PRINT"|                                      |"
3520 PRINT"| Número de opción...                  |"
3530 PRINT"└─────────────────────────────────────┘"
3540 A$=INKEY$
3550 IF A$=""THEN 3540
3560 BEEP
3570 IF A$<"1" OR A$>"6"THEN 3550
3580 ON VAL(A$) GOTO 3590,3650,3710,3770,3830,1480
3590 CLS
3600 PRINT"P(W)=V*I":PRINT:PRINT
3610 INPUT"TENSION ";V:PRINT
3620 INPUT"INTENSIDAD ";I:PRINT
3630 PRINT:PRINT"POTENCIA=";CSNG(V*I);"WATIOS."
3640 GOTO 3890
3650 CLS
3660 PRINT"P(W)=R*I²":PRINT:PRINT
3670 INPUT"RESISTENCIA: ";R:PRINT
3680 INPUT"INTENSIDAD ";I:PRINT
3690 PRINT:PRINT"POTENCIA="CSNG(R*I^2);"WATIOS."
3700 GOTO 3890
3710 CLS
3720 PRINT"P(W)=V²/R":PRINT:PRINT
3730 INPUT" TENSION: ";V:PRINT
3740 INPUT" RESISTENCIA: ";R:PRINT
3750 PRINT:PRINT"POTENCIA=";CSNG(V^2/R);"WATIOS."
3760 GOTO 3890
3770 CLS
3780 PRINT"Imax=SQR(P/R)":PRINT
3790 INPUT"POTENCIA: ";P:PRINT
3800 INPUT"RESISTENCIA: ";R:PRINT
3810 PRINT"INTENSIDAD MAX.=";CSNG(SQR(P/R));"AMPERIOS."
3820 GOTO 3890
```

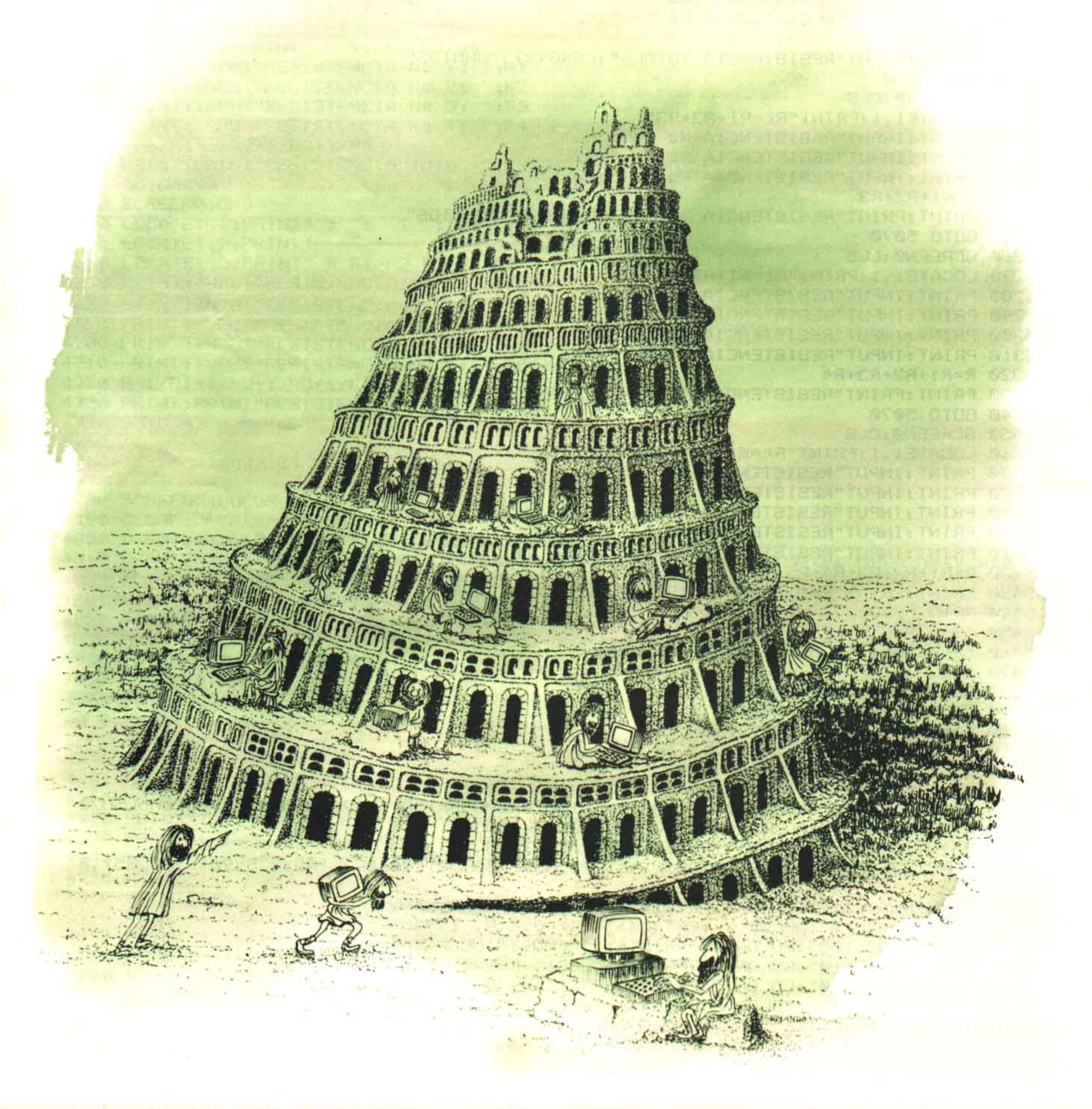
```
3830 CLS .
3840 PRINT"Vmax=SQR(P*R)":PRINT
3850 INPUT"POTENCIA: ";P:PRINT
3860 INPUT"RESISTENCIA: ";R:PRINT
3870 PRINT"TENSION MAX.=";CSNG(SQR(P*R));"VOLTIOS."
3880 GOTO 3890
3890 LOCATE 0,18:PRINT"Pulsa una tecla para volver al menú."
3900 IF INKEY$="" THEN 3900
3910 BEEP
3920 GOTO 3350
3930 '
3940 '---comienza la opción nº 5---
3950 '
3960 SCREEN2:COLOR 5,1,1:CLS
3970 CLOSE:OPEN"GRP:"AS#1
3980 DRAW"BM0,10":PRINT#1,"  -RESISTENCIAS EN PARALELO-
3990 DRAW"BM30,113R20U20R10U2R10D4L10U2BR10R10D10ND10L10U2L10D2NL10D2R10U2BD10U2
L10D2NL10D2R10U2R10NR20D10L10U2L10D2NL10D2R10U2R10D10L10U2L10D4R10U2BL10L10U20":
PAINT(65,93),5:PAINT(65,103),5:PAINT(65,113),5:PAINT(65,123),5:PAINT(65,133),5
4000 DRAW"BM110,100":COLOR2:PRINT#1," 1  1  1  1      1"
4010 DRAW"BM110,110":PRINT#1,"──=──+──+──+...──"
4020 DRAW"bm110,120":PRINT#1,"Rt R1 R2 R3     Rn"
4030 DRAW"BM5,185":COLOR5:PRINT#1,"Pulsa una tecla para continuar."
4040 IF INKEY$=""THEN4040
4050 SCREEN0:WIDTH40:CLS
4060 PRINT"┌──────────────────────────────────────┐"
4070 PRINT"|    ---RESISTENCIAS EN PARALELO---    |"
4080 PRINT"├──────────────────────────────────────┤"
4090 PRINT"|                                      |"
4100 PRINT"|   Para poder acceder a los cálculos  |"
4110 PRINT"| de las resistencias intercaladas en  |"
4120 PRINT"| un montaje en paralelo debes de in-  |"
4130 PRINT"| troducir el número de resistencias   |"
4140 PRINT"| a calcular, y seguidamente ir intro- |"
4150 PRINT"| duciendo los datos conforme los vaya |"
4160 PRINT"| solicitando el ordenador.            |"
4170 PRINT"|                                      |"
4180 PRINT"| (El número mínimo de resistencias a  |"
4190 PRINT"| calcular es de 2 y el máximo es 5)   |"
4200 PRINT"|                                      |"
4210 PRINT"| Para retornar al menú principal no   |"
4220 PRINT"| tienes más que introducir el nº 1    |""
4230 PRINT"|                                      |"
4240 PRINT"└──────────────────────────────────────┘"
4250 C$=INKEY$:IFC$=""THEN4250
4260 BEEP
4270 IFC$<"1"ORC$>"5"THEN 4250
4280 ON VAL(C$) GOTO 1480,4330,4420,4520,4630
4290 LOCATE0,20:PRINT" Pulsa una tecla para volver al  menú."
4300 IF INKEY$=""THEN 4300
4310 BEEP
4320 GOTO 4050
4330 SCREEN0:CLS
4340 LOCATE1,1:PRINT" 1  1  1"
4350 LOCATE1,2:PRINT"──=──+── "
4360 LOCATE1,3:PRINT" R R1 R2"
4370 PRINT:INPUT"RESISTENCIA Nº 1: ";R1
4380 PRINT:INPUT"RESISTENCIA Nº 2: ";R2
4390 R=1/R1+1/R2
4400 PRINT:PRINT"RESISTENCIA TOTAL=";CSNG(1/R);"OHMIOS"
4410 GOTO 4290
4420 SCREEN0:CLS
4430 LOCATE1,1:PRINT" 1  1  1  1"
```


GANE
7.000 PTAS.
todos los meses
PARTICIPANDO EN NUESTRO CONCURSO
A partir del próximo número MSX premiará mensualmente los programas que hagan llegar los lectores.
Para participar en este concurso abierto, todo aficionado a los ordenadores con este estándar deberá hacer llegar a la redacción de la revista el listado, un cassette y un texto explicativo.
Entre todos los programas que recibamos cada mes, serán seleccionados para su publicación aquellos que reúnan los siguientes criterios:
— Originalidad de la aplicación.
— Simplicidad del método de programación.
La única condición para participar en el concurso será que los programas no hayan sido publicados previamente en ninguna revista.

```
4440 LOCATE1,2:PRINT"──═──┼──┼──"
4450 LOCATE1,3:PRINT" R R1 R2 R3"
4460 PRINT:INPUT"RESISTENCIA Nº 1: ";R1
4470 PRINT:INPUT"RESISTENCIA Nº 2: ";R2
4480 PRINT:INPUT"RESISTENCIA Nº 3: ";R3
4490 R=1/R1+1/R2+1/R3
4500 PRINT:PRINT"RESISTENCIA TOTAL=";CSNG(1/R);"OHMIOS"
4510 GOTO4290
4520 SCREEN0:CLS
4530 LOCATE1,1:PRINT" 1  1  1  1  1"
4540 LOCATE1,2:PRINT"──═───┼───┼───┼──"
4550 LOCATE1,3:PRINT" R R1 R2 R3 R4"
4560 PRINT:INPUT"RESISTENCIA Nº 1: ";R1
4570 PRINT:INPUT"RESISTENCIA Nº 2: ";R2
4580 PRINT:INPUT"RESISTENCIA Nº 3: ";R3
4590 PRINT:INPUT"RESISTENCIA Nº 4: ";R4
4600 R=1/R1+1/R2+1/R3+1/R4
4610 PRINT:PRINT"RESISTENCIA TOTAL=";CSNG(1/R);"OHMIOS"
4620 GOTO4290
4630 SCREEN0:CLS
4640 LOCATE1,1:PRINT" 1  1  1  1  1  1"
4650 LOCATE1,2:PRINT"──═───┼───┼───┼───┼──"
4660 LOCATE1,3:PRINT" R R1 R2 R3 R4 R5"
4670 PRINT:INPUT"RESISTENCIA Nº 1: ";R1
4680 PRINT:INPUT"RESISTENCIA Nº 2: ";R2
4690 PRINT:INPUT"RESISTENCIA Nº 3: ";R3
4700 PRINT:INPUT"RESISTENCIA Nº 4: ";R4
4710 PRINT:INPUT"RESISTENCIA Nº 5: ";R5
4720 R=1/R1+1/R2+1/R3+1/R4+1/R5
4730 PRINT:PRINT"RESISTENCIA TOTAL=";CSNG(1/R);"OHMIOS"
4740 GOTO 4290
4750 '
4760 '---comienza la opción nº 5---
4770 '
4780 SCREEN2:COLOR 5,1,1:CLS
4790 CLOSE:OPEN"grp:"AS#1
4800 DRAW"bm15,10":PRINT#1,"  ──RESISTENCIAS EN SERIE──"
4810 DRAW"BM22,110R10U3R10D3R5U3R10D3R5U3R10D3R5U3R10D3R5U3R10D3NR10D3L10U3L5D3L
10U3L5D3L10U3L5D3L10U3L5D3L10U3":PAINT(37,110),5:PAINT(52,110),5:PAINT(67,110),5
:PAINT(82,110),5:PAINT(97,110),5
4820 DRAW"BM120,107":COLOR2:PRINT#1,"Rt=R1+R2+R3+...Rn"
4830 DRAW"BM5,185":COLOR5:PRINT#1,"Pulsa una tecla para continuar."
4840 IF INKEY$=""THEN4840
4850 SCREEN0:WIDTH40:CLS
4860 PRINT"┌────────────────────────────────────┐"
4870 PRINT"|     ───RESISTENCIAS EN SERIE───     |"
4880 PRINT"├────────────────────────────────────┤"
4890 PRINT"|                                    |"
4900 PRINT"|   Para calcular la resistencia to- |"
4910 PRINT"| tal, no tienes más que introducir  |"
4920 PRINT"| el número de resistencias a calcu- |"
4930 PRINT"| lar, e ir introduciendo los datos  |"
4940 PRINT"| conforme los vaya pidiendo el orde-|"
4950 PRINT"| nador.                             |"
4960 PRINT"|                                    |"
4970 PRINT"| (El número mínimo de resistencias a|"
4980 PRINT"| calcular es 2 y el máximo es de 5).|"
4990 PRINT"|                                    |"
5000 PRINT"|  Para poner fin al programa, intro-|"
5010 PRINT"| cir el número 1.                   |"
5020 PRINT"└────────────────────────────────────┘"
5030 Z$=INKEY$:IFZ$=""THEN5030
5040 BEEP
```

```
5050 IFZ$<"1"ORZ$>"5"THEN5030
5060 ON VAL(Z$) GOTO 5480,5110,5180,5260,5350
5070 LOCATE0,20:PRINT"Pulsa una tecla para volver al menú."
5080 IF INKEY$=""THEN5080
5090 BEEP
5100 GOTO 4850
5110 SCREEN0:CLS
5120 LOCATE1,1:PRINT"Rt=R1+R2"
5130 PRINT:INPUT"RESISTENCIA Nº 1: ";R1
5140 PRINT:INPUT"RESISTENCIA Nº 2: ";R2
5150 R=R1+R2
5160 PRINT:PRINT"RESISTENCIA TOTAL=";CSNG(R);"OHMIOS"
5170 GOTO5070
5180 SCREEN0:CLS
5190 LOCATE1,1:PRINT"Rt=R1+R2+R3"
5200 PRINT:INPUT"RESISTENCIA Nº 1: ";R1
5210 PRINT:INPUT"RESISTENCIA Nº 2: ";R2
5220 PRINT:INPUT"RESISTENCIA Nº 3: ";R3
5230 R=R1+R2+R3
5240 PRINT:PRINT"RESISTENCIA TOTAL=";CSNG(R);"OHMIOS"
5250 GOTO 5070
5260 SCREEN0:CLS
5270 LOCATE1,1:PRINT"Rt=R1+R2+R3+R4"
5280 PRINT:INPUT"RESISTENCIA Nº 1: ";R1
5290 PRINT:INPUT"RESISTENCIA Nº 2: ";R2
5300 PRINT:INPUT"RESISTENCIA Nº 3: ";R3
5310 PRINT:INPUT"RESISTENCIA Nº 4: ";R4
5320 R=R1+R2+R3+R4
5330 PRINT:PRINT"RESISTENCIA TOTAL=";CSNG(R);"OHMIOS"
5340 GOTO 5070
5350 SCREEN0:CLS
5360 LOCATE1,1:PRINT"Rt=R1+R2+R3+R4+R5"
5370 PRINT:INPUT"RESISTENCIA Nº 1: ";R1
5380 PRINT:INPUT"RESISTENCIA Nº 2: ";R2
5390 PRINT:INPUT"RESISTENCIA Nº 3: ";R3
5400 PRINT:INPUT"RESISTENCIA Nº 4: ";R4
5410 PRINT:INPUT"RESISTENCIA Nº 5: ";R5
5420 R=R1+R2+R3+R4+R5
5430 PRINT:PRINT"RESISTENCIA TOTAL=";CSNG(R);"OHMIOS"
5440 GOTO 5070
5450 '
5460 '---pantalla de fin de programa---
5470 '
5480 COLOR 5,1,1:SCREEN2
5490 CLOSE:OPEN"GRP:"AS#1
5500 LINE(255,191)-(0,0),2,BF
5510 LINE(245,181)-(10,10),15,BF
5520 LINE(235,171)-(20,20),2,BF
5530 LINE(225,161)-(30,30),1,BF
5540 DRAW"BM60,40C2NR40D30R10U10R15U5L15U10R30U5BR15C15NR20D5R5D20L5D5R20U5L5U20
R5U5BR15C2R10M185,60U20R10D30L10M160,50D20L10U30"
5550 PAINT(65,55),2:PAINT(125,55),15:PAINT(173,55),2
5560 LINE(225,80)-(30,75),8,BF:LINE(225,85)-(30,80),10,BF:LINE(225,90)-(30,85),8
,BF
5570 DRAW"bm57,11":COLOR 1:PRINT#1,"[redacted]"
5580 DRAW"bm67,11":COLOR 2 :PRINT#1,"(C) D.C.R. 1.985"
5590 DRAW"BM67,100":COLOR 15:PRINT#1,"MEMORIA OCUPADA:"
5600 DRAW"BM60,110":COLOR 13:PRINT#1,28815-FRE(0);"BYTES."
5610 DRAW"BM90,130":COLOR 5:PRINT#1,"Pulsar -M- "
5620 DRAW"BM50,145":COLOR 5:PRINT#1,"para volver al menú."
5630 T$=INKEY$:IF T$=""THEN5630
5640 IFT$="M" OR T$="m" THEN RUN 1480 ELSE 5650
5650 SCREEN1:WIDTH32:COLOR 5,1,1:CLS:KEY ON
```

Las torres de Hanoi



LOS JUEGOS ELECTRONICOS

En este programa se plantea la resolución de un antiguo problema matemático, consistente en trasladar todos los discos de la torre número 1 hasta la tercera torre en el menor número de intentos y en un orden establecido. El orden viene marcado por la cualidad de que los discos se han de colocar de mayor a menor. Para ello, iremos llevando uno a uno de torre en torre, siempre y cuando el mayor esté por debajo del menor.

El juego contiene las instrucciones de uso, y ofrerce la posibilidad de elegir el número de torres, a emplear hasta un máximo de 6. El número de movimientos que permite el programa es ilimitado, aunque, lógicamente hay que intentar realizarlo en el menor número de veces posible.

La presentación y el juego en sí, está bien realizado. Se trata de un programa muy entretenido y especialmente recomendado para las mentes matemáticas y calculadoras.

Diego Castaño Rodríguez
Sevilla

```
10 '-----------TORRES DE HANOI----------
20 '-----(C) Diego Castaño Rodriguez----
30 '-------Dos Hermanas (Sevilla)-------
40 '--------- Para MSX MAGAZINE---------
50 '
60 GOSUB 810
70 SCREEN0:WIDTH39:KEY OFF:COLOR 5,1,1
100 '--reglas del juego--
110 CLS
120 LOCATE 10,0:PRINT "TORRES DE HANOI"
130 LOCATE 10,1:PRINT "====== == =====":PRINT
140 LOCATE 10,3:PRINT " INSTRUCCIONES"
150 LOCATE 10,4:PRINT " -------------"
160 PRINT:PRINT" -1- El juego consiste en trasladar "
```

```
170 PRINT "una serie de discos desde la torre  1"
180 PRINT "a la torre 2."
190 PRINT:PRINT " -2- Hay que tener en cuenta que no se"
200 PRINT "puede colocar un disco mayor encima de"
210 PRINT "otro más pequeño."
220 PRINT:PRINT " -3- Hay que finalizar el juego con el"
230 PRINT "menor número posible de movimientos."
240 PRINT:PRINT " -4- El juego dispone de 1 a 6 discos."
250 LOCATE 4.22:PRINT"Pulsa una tecla para continuar."
260 IF INKEY$=""THEN260
270 BEEP
280 CLS
285 DIM T(7.3).A(3)
286 D$="[redacted]"
290 INPUT "Número de discos a mover (1÷6): ";AL
300 BEEP
310 IF AL>6 THEN PRINT "Son demasiados.":GOTO290
320 '--inicializacion de las torres--
330 NM=1
340 FORI=1TOAL
350 T(I.1)=AL-I+1
360 T(I.2)=0
370 T(I.3)=0
380 NEXT
390 A(1)=AL:A(2)=0:A(3)=0
400 '--presentacion de las torres--
410 CLS
420 LOCATE 11.0:PRINT "TORRES DE HANOI"
430 LOCATE 11.1:PRINT "====== == =====":PRINT :PRINT
440 FOR AP=AL TO 1 STEP -1
450 PRINT TAB(6)"■";TAB(18)"■";TAB(30)"■"
460 FORI=1TO3
470 PRINT TAB(I*12-6-T(AP.I))LEFT$(D$.T(AP.I)*2+1);
480 NEXTI
490 PRINT
500 NEXTAP
510 PRINT TAB(5)"(1)";TAB(17)"(2)";TAB(29)"(3)"
520 PRINT  D$;D$;D$
530 PRINT
540 '--introduccion de datos de los moimientos--
550 PRINT
560 IFA(1)=0ANDA(2)=0 THEN 710
570 PRINT "Intentos:";NM
580 INPUT "Torre de salida ";O
590 BEEP
600 IFA(O)=0THENPRINT "La torre";O;"está vacia.":GOTO540
610 INPUT "Torre de llegada ";F
620 BEEP
630 IFA(F)=0THEN650
640 IFT(A(F).F)<T(A(O).O)THENPRINT "Ese movimiento no es lógico.":GOTO540
650 NM=NM+1
660 A(F)=A(F)+1
670 T(A(F).F)=T(A(O).O)
680 T(A(O).O)=0
690 A(O)=A(O)-1
700 GOTO400
710 '--juego terminado--
720 PRINT "Enhorabuena. lograste pasar los discos"
730 PRINT "en";NM-1;"movimientos.":PRINT
740 FORT=1TO2500:NEXT
750 INPUT "QUIERES SEGUIR JUGANDO (S/N) ";J$
760 BEEP
770 IF J$="S" OR J$="s" THEN RUN 280 ELSE 780
```

```
780 SCREEN1:COLOR 15.4.7
790 LOCATE 1.10:PRINT "Memoria Ocupada";28815-FRE(0);"Bytes."
800 END
810 '
820 '--cabecera de presentacion--
830 '
840 SCREEN2:COLOR 15.4.7:CLS
850 OPEN"GRP:"AS#1
860 DRAW"BM40.10":COLOR1:PRINT#1."[redacted]"
870 DRAW"BM50.10":COLOR 5:PRINT#1."LAS TORRES DE HANOI"
880 D$="R5D5R5D5L5D5R10D5L10D5R15D5L15D5R20D5L20D5R25D5L55U5R25U5L20U5R20U5L15U5
R15U5L10U5R10U5L5U5R5U5R2
890 A=4:B=3:C=2:K=15:L=1:M=13
900 DRAW"BM35.105;S=A;C=K:XD$;":PAINT(36.106).15:DRAW"BM80.80;S=B;C=L;XD$;":PAIN
T(81.81).1:DRAW"BM120.110;S=C;C=M:XD$;":PAINT(121.111).13
910 E$="r14u3112u2r12u319u2r9u10r2d10r9d219d3r12d2112d3r28u317u2r7u13r2d13r7d217
d3r28u315u2r5u13r2d13r5d215d3r14d2190u2"
920 DRAW"bm120.90;s=a;c=1:xe$;":PAINT(121.91).1:DRAW"bm30.50;s=b;c=m:xe$;"
930 DRAW"BM10.180":COLOR 1:PRINT#1,"[redacted]"
940 DRAW"BM20.180":COLOR 5:PRINT#1."(c) Diego Castaño. 1985-SE "
950 A$="V12T20006L8EDL4C05A06DFEL8DC05L4B06L8EDL4CL805BAL2G#A"
960 C$="V13T200R203L8AGL4FDEF#G#EADEEL2A"
970 F$="V14T20005L8C04B05L4CL8DEL4FL804BAL4BL805CDL4E04L8AGL4A05L8DCL204B05C"
980 PLAYA$
990 PLAYA$.C$
1000 PLAYA$.C$.F$
1010 FORDE=1TO2500:NEXTDE
1020 RETURN
1030 '----CSAVE"HANOI"-------
```



Para usuarios de Reus y comarca, se ha formado en Reus el club «Todo MSX», si deseas apuntarte o recibir información escríbenos a la siguiente dirección: Juan Carlos de la Llana. C/ Llenoa, 12. Reus (Tarragona). Indicando en el sobre que es para el club «Todo MSX».

Intercambio programas MSX. Escribir a F. Escribano Zamorano. Avda. 1 de Mayo, 11. La Llagosta (Barcelona).

Deseo contactar con algún club o usuarios de MSX. También compro programas. Escribir a José Manuel García. C/ Ermita, 8. Ronda (Málaga).

Intercambio programas. Los interesados escribir a David Martí y Rubio. C/ José Martínez González, 75. Elda (Alicante).

Intercambio programas. Contactar con Miguel Angel Yáñez Camacho. C/ Perú, 21. 41012 Sevilla, o llamar al Tel.: (954) 61 26 36.

Deseo cambiar programas MSX. Escribir a Urb. Los Claveles, Mirena de Alcor (Sevilla) o llamar al Tel.: (954) 74 21 85, preguntar por José M. Martínez Mateos.

Cambio o vendo programas (exclusivamente en disco), tanto en formato de 3,5 pulgadas como de 5 1/4. Contactar con Miguel González. C/ Héroes de Baleares, 2. 39010 Santander, o llamar al Tel.: (942) 34 16 87.



Vendo Spectravideo SVI-328 con grabadora, dos joysticks, tabla gráfica y 18 juegos. Precio a convenir. Dirigirse a Antonio Muñoz Glez. C/ Juan Ramón Jiménez, 43. 35011 Las Palmas, o llamar al Tel.: (928) 20 16 59.

Vendo ordenador Sony 64K con más de 70 programas diversos, por 73.000 ptas. Escribir a Jorge Lafuente Bartra, Avda. de Madrid, 30-34. 08028 Barcelona, o llamar al Tel.: (93) 333 45 36.

Vendo ordenador Sony HB-55P con cartucho para expansión de memoria de 16K, un cassette Fujiyama, con joystick Quickshot I y más de 40 programas, todo por 60.000 ptas. Los interesados pueden llamar al Tel.: (986) 20 93 37, preguntar por Javier.

SERVICIO DE EJEMPLARES ATRASADOS

ESTOS SON LOS EJEMPLARES DE MSX MAGAZINE APARECIDOS EN EL MERCADO CON UN RESUMEN DE SU CONTENIDO



Núm. 1
¿Qué es el MSX? Su BASIC, periféricos, programas, software.



Núm. 2
Generación de sonido, MSX-DOS, el ordenador por dentro, programas, noticias.



Núm. 3
Los joysticks, 256 caracteres programables, Z80 corazón de león, compro/vendo/cambio.



Núm. 4
Las comunicaciones entre ordenadores, la jerga informática, trucos, rincón del lector.



Núm. 5
Comandos de entrada/salida, el BASIC MSX comparado con Spectrum y Commodore 64, Código Máquina.



Núm. 6
Los 8 magnificos (test gigante), el bus de expansión, los misterios de la grabación, programas.



Núm. 7
Analizamos el Generador de Sonido. Aplicaciones matemáticas con el ordenador. La memoria de video. Trucos, noticias.



Núm. 8
Compact Disc, el periférico del futuro. Test: Dynadata DPC-200. Continuamos con la memoria de video. Libros, software, programas, trucos.



Núm. 9
Características técnicas del Compact Disc. Tratamiento de datos. Test: Quick Disk. Trucos, libros, noticias, programas.

PARA HACER SU PEDIDO, RELLENE ESTE CUPON, HOY MISMO Y ENVIELO A MSX MAGAZINE BRAVO MURILLO, 377. Tel. 733 96 62 - 28020 MADRID

Ruego me envíen los siguientes números atrasados ....

al precio de **250** ptas. cada uno. Cuyo importe abonaré:

☐ POR CHEQUE ☐ CONTRA REEMBOLSO ☐ CON MI TARJETA DE CREDITO

☐ AMERICAN EXPRESS ☐ VISA ☐ INTERBANK

Número de mi tarjeta ....

Fecha de caducidad ....

NOMBRE ....

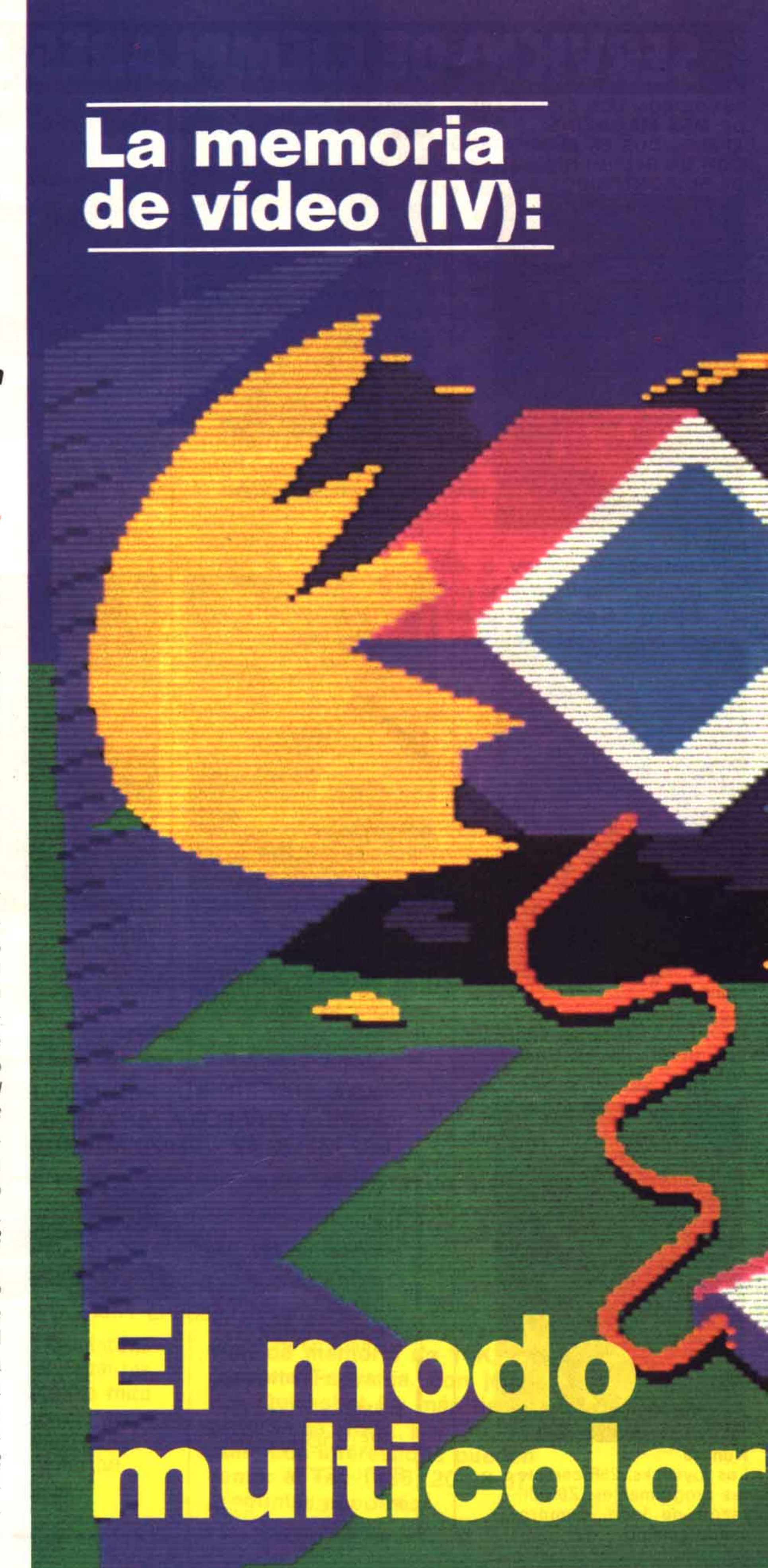
DIRECCION ....

POBLACION .... C.P. ....

PROVINCIA ....

La memoria de vídeo (IV):

El modo multicolor



El modo de gráficos SCREEN 3 tiene menos definición que el modo SCREEN 2, pero posee otras ventajas muy útiles para el programador. En este número os ayudamos a descubrirlas.

Una de las pocas desventajas que posee el modo de gráficos *SCREEN 2* es, como explicábamos en el número anterior, que sólo acepta un color de tinta y uno de fondo en cada agrupación de 8 *pixels*, y por ello sucedían accidentes como éste:

```
10 COLOR 4,15,15: SCREEN 2
20 CIRCLE (100,100),50,4,,,1.4
30 CIRCLE (100,100),60,8,,,1.4
40 GOTO 40
```

Cuando este programa funcione, serán trazadas en la pantalla dos circunferencias concéntricas, una azul y otra roja. Debido a su proximidad y a que se trazan en *SCREEN 2*, una parte de la circunferencia interior se impregnará de rojo. Para evitar esto, es necesario sustituir en la orden 10 «*SCREEN 2*» por «*SCREEN 3*». En *SCREEN 3* las dos circunferencias aparecerán con sus colores, pero con un trazado mucho más grueso. Esto es debido a una característica muy peculiar del modo *SCREEN 3* que ahora resumiremos.

Mientras que en el modo *SCREEN 2* todas las órdenes de gráficos se refieren a *pixels*, en el modo *SCREEN 3* se refiere a bloques de 4 x 4 *pixels*. De esta manera, las órdenes que hacen que en *SCREEN 2* un *pixel* se impregne de un color, en *SCREEN 3* logran que ese color impregne todo el bloque dentro del que se ha-

lla el *pixel* referido. Como consecuencia de esta característica, todos los dibujos trazados en *SCREEN 3* aparecen excesivamente gruesos, y puede dar la impresión de que este modo de pantalla no se debe utilizar, o que se debe utilizar lo mínimo, cuando lo cierto es que es utilísimo para el programador en muchos de sus aspectos.

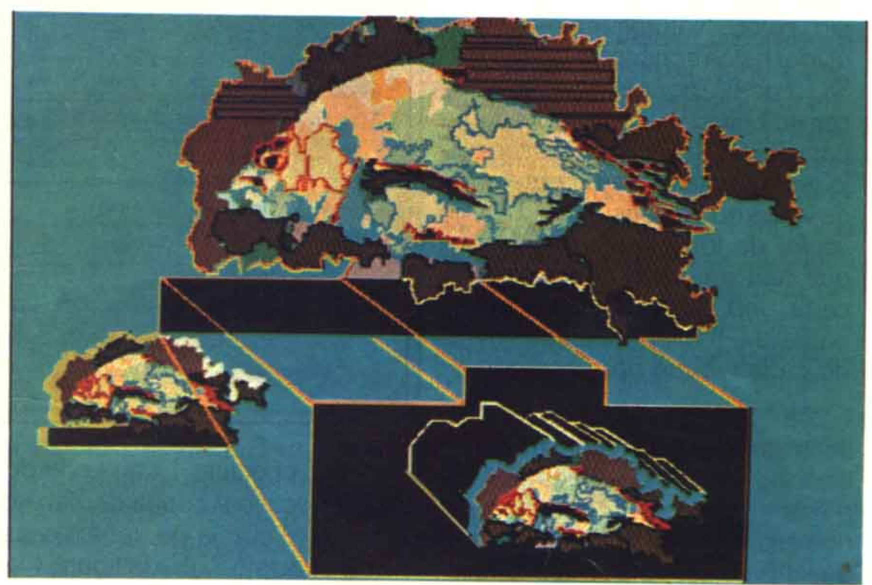
Una de sus utilidades es el trazado de rótulos. Como sabéis, para imprimir caracteres en un modo de gráficos es necesario abrir un canal con la orden *OPEN «GRP:» FOR OUTPUT AS #1*, para a continuación introducir mediante un *PRINT # 1* el texto que queráis imprimir. En el modo *SCREEN 2* los caracteres se imprimen del mismo tamaño que en el modo *SCREEN 1*, pero en *SCREEN 3* se imprimen mucho más grandes. Para observar el tamaño, introducid este programa:

```
10 OPEN «GRP:» FOR OUTPUT
   AS #1
20 SCREEN 3
30 PRESET (80,80):PRINT # 1,
   «MSX»
40 GOTO 40
```

Utilizamos la instrucción *PRESET* (80,80) para situar los caracteres en la zona central de la pantalla, pues, de esa manera, la primera letra del texto (la «M», en este caso) se imprime en un cuadrado de 32 x 32 *pixels* cuyo vértice superior izquierdo es el *pixel* (80,80).

La tabla 17: el mapa de la pantalla

El modo *SCREEN 3* dispone de cuatro tablas en la memoria de vídeo para el almacenamiento de datos: las tablas 15, 17, 18, y 19. Las tablas 18 y 19 serán tratadas en el número siguiente, pues su función es almacenar los datos de



las figuras móviles, los famosos *«sprites»*.

Vamos a hablar primero de la tabla 17, pues consideramos que el aprendizaje de la tabla 15 entraña mayor dificultad.

La tabla 17 tiene una extensión de 1536 octetos, y en nuestros ordenadores MSX comienza en el octeto 0. (Como siempre, para averiguar dónde comienza la tabla 17 en vuestro ordenador de-

béis introducir *PRINT BASE* (17).) Si recordáis las tablas del modo *SCREEN 2,* recordaréis que la tabla 12 era el mapa de la pantalla, y la tabla 11 estaba destinada a almacenar los colores de ese mapa de pantalla. Pues bien, la tabla 17 del sistema de almacenamiento del modo *SCREEN 3,* cumple las funciones de ambas tablas: es un mapa de pantalla que almacena los colores. ¿Y por qué el modo *SCREEN 3* necesita sólo una tabla para lo que el modo *SCREEN 2* requiere dos? La respuesta es sencilla: el modo *SCREEN 3* deja un poco de lado los conceptos de «color de tinta» y «color de fondo» (conceptos de los que se basa la tabla 12) y considera que cada bloque de 4 x 4 *pixels* tiene su propio color. Naturalmente, podéis utilizar estos conceptos desde *BASIC.* Podéis, por ejemplo, establecer un color de fondo rojo en una pantalla *SCREEN 3,* y todo será rojo en la pantalla, pero la memoria de vídeo no ha considerado al rojo «color de fondo», sino «color del que han de impregnarse todos y cada uno de los bloques de la pantalla», habiéndose considerado cada bloque por separado.

Los 1536 octetos de la tabla 17 se organizan de la siguiente manera: agrupan los bloques de 4 x 4 *pixels* en grupos de dos bloques que estén uno al lado del otro (es decir, parejas de bloques alineados horizontalmente). Los octetos de la tabla 17 contendrán un número entre 0 y 255, cuya primera cifra en hexadecimal será el color del bloque situado a la izquierda, y la segunda cifra el color del bloque situado a la derecha.

Cada octeto se corresponde, pues, con una pareja de bloques. El octeto 0 se corresponde con la pareja de bloques situada en la esquina superior izquierda de la pantalla, el octeto 1 con la de debajo, el octeto 2 con la de debajo, el octeto 3 con la de debajo, y así hasta el octeto 7. (En la figura 1 se puede observar la distribución de los octetos, en la que cada rectángulo es una pareja de bloques.) El octeto 8 corresponde a la pareja situada a la derecha de la pareja del octeto 0, y siguiendo esta sucesión en vertical, la del octeto d15 se sitúa a la derecha de la del octeto 7. Lo mismo sucede con las parejas de los octetos 16-23, 24-31, etc. La pareja del octeto 255 estará, por lo tanto, en la misma línea horizontal que la del octeto 7, situada en el extremo derecho de la pantalla.

A este conjunto de 256 parejas alineadas en un rectángulo de 8 parejas de lado y 32 parejas de base le vamos a llamar «franja» de la pantalla. Existen un total de 6 franjas en la pantalla, que se rellenan de igual forma que la primera, correspondiendo la paréja de la esquina inferior derecha (la última pareja de la última franja) al octeto 1535, completando con éste los 1536 octetos de la tabla 17.

Si queréis saber qué octeto debéis cambiar para impregnar de color un bloque en el que se halla un *pixel* de coordenadas conocidas (x,y), debéis aplicar esta fórmula:

Primer octeto de la tabla + 256 x *INT*(y/32) + 8 x *INT*(x/8) + *INT*((y*MOD*32)/4)

Si el resultado de la fórmula siguiente es 0, el *pixel* está en el bloque de la izquierda de la pareja correspondiente, y si es 1, está en el de la derecha:

INT((x*MOD*8)/4)

ZONA DE BORDE

ZONA DE BORDE

0	8	16	24
1	9	17	25
2	10	18	26
3	11	19	27
4	12	20	28
5	13	21	29
6	14	22	
7	15	23	
256	264		
257	265		
258			
259			

PRIMER CAMPO

SEGUNDO CAMPO

La tabla 17 en la práctica

Comencemos con la clásica aplicación del efecto «*FLASH*». Teclead este programa en vuestro ordenador:

```
10 SCREEN 3
20 DATA 44,4F,F4,FF,F4,F4,44,
   44,F4,4F,44,FF,44,44,44,44,
   44,44,F4,F4,F4,F4,44,44
30 DATA FF,F4,4F,44,4F,4F,FF,
   FF,4F,F4,FF,44,FF,FF,FF,FF,
   FF,FF,4F,4F,4F,4F,FF,FF
40 RESTORE 20:FOR T=648
   TO 671:READ Q$:VPOKE T,
   VAL («&H» + Q$):NEXT
50 FOR R=1 TO 200:NEXT
60 FOR T=648 TO 671:READ
   Q$:VPOKE T, VAL(«&H» +
   Q$):NEST
70 FOR R=1 TO 200:NEXT
80 GOTO 40
```

Este programa emula el efecto «*FLASH*» en una zona central de la pantalla (la situada entre los octetos 648 y 671) con una «A» mayúscula. Para ello ha sido necesario digitalizar el patrón de la «A», y digitalizar también este mismo patrón en negativo. (Esto último es muy fácil tomando como base el patrón de la «A». Como en el patrón hemos supuesto el color blanco *(&HF)* como el color que en un modo de pantalla de texto sería el «color de tinta» y el color azul *(&H4)* como lo que sería «color de fondo», para conseguir el patrón de la «A» en negativo digitalizado no hay más que tomar el patrón de la «A», y situar «F» donde ponía «4», y «4» donde ponía «F». Si os fijáis, esa es la diferencia que existe entre los *DATA* de la línea 20 y 30.) El patrón de la «A» en positivo está en el *DATA* de la orden 20, y el de «A» en negativo en la orden 30. El resto del programa es muy sencillo: los bucles de variable *T* leen los datos y los sitúan en la pantalla con las órdenes *VPOKE*, y los bucles de variable *R* retardan el proceso. La orden «*RESTORE 20*» establece que el *DATA* que comienza a leer la orden 40 ha de ser el situado en la línea 20. Si esta expresión no hubiera sido incluida, en el momento en que el ordenador hubiera vuelto a la orden 40 por medio de la línea 80 se habría cometido un error «*Out of DATA in 40*».

Para textos mayores no es aconsejable el efecto «*FLASH*» debido a su lentitud, pero sí es útil utilizarlo en letras sueltas o en pequeños rectángulos. Por ejemplo, si en un juego esquemático (un juego hecho sin precisión en los dibujos, para lo cual es ideal el modo *SCREEN 3)* queréis situar en



la pantalla algún tipo de «diablillo», o «marcianito» y queréis resaltar que está nervioso, o que se puede mover muy deprisa, o que constituye algo importante dentro del desarrollo del juego, podéis establecer un bucle para pintarlo y hacerlo desaparecer instantáneamente, sin bucle retardador, con las dos órdenes de pintarlo y borrarlo seguidas, una tras la otra.

La tabla 15

La tabla 15 del sistema de almacenamiento del modo *SCREEN 3* posee un total de 768 octetos, situados desde el 2048 al 2815 en la memoria de vídeo de nuestros ordenadores MSX. Su función es similar a la de la tabla 10 en el sistema de almacenamiento del modo *SCREEN 2*, es decir, su objetivo es situar unos cuadros de la pantalla en función de otros que pertenezcan a su mismo campo de pantalla.

La diferencia principal entre la tabla 10 y la tabla 15 es que mientras que la tabla 10 dividía la pantalla en tres zonas o «campos» claramente diferenciados, la tabla 15 divide la pantalla en cuatro zonas, cada una de las cuales tiene seis líneas de cuadros. Pero estas seis líneas de cada zona NO ESTAN UNA INMEDIATAMENTE DEBAJO DE OTRA, sino que ESTAN ENTREMEZCLADAS. El resultado final es el siguiente: la primera línea de la pantalla empezando por arriba es la primera línea de la primera zona, la segunda es la primera de la segunda zona, la tercera es la primera de la tercera zona y la cuarta es la primera de la cuarta zona. Pero la quinta ya es la segunda de la primera zona, la sexta es la segunda de la segunda zona, y así hasta el final de la pantalla. (NOTA: Cuando nos referimos a líneas de cuadros nos referimos a cuadros de los usados en *SCREEN 1* y en *SCREEN 2*, es decir, cuadros de 8 x 8 *pixels*, que en el modo *SCREEN 3* contienen 2 x 2 bloques de los usados en este modo de gráficos). De esta manera, la última línea de la pantalla, la línea vigésimo cuarta, será la sexta línea de la cuarta zona.

Cada octeto de la tabla 15 hace referencia a un cuadro de la pantalla (4 zonas x 6 líneas x 32 cuadros por línea = 768 cuadros ≐ 768 octetos) y el número que contiene cada octeto indica en función de qué cuadro de su zona se

ha de pintar el cuadro correspondiente a ese octeto. Lógicamente, los octetos deL 2048 al 2079 se ocuparán de la primera línea del primer campo, los del 2080 al 2111 de la primera línea del segundo campo, y así, en intervalos de 32 octetos, siguiendo el orden de líneas de la pantalla de arriba a abajo, los octetos del 2784 al 2815 se ocuparán de la sexta línea de la cuarta zona.

La desventaja que posee esta tabla, al igual que la tabla 10, es que sólo podéis poner un cuadro en función de otro que esté forzosamente en su misma zona. Si queréis saber a qué zona de las cuatro (numeradas de 0 a 3) pertenece un *pixel* de coordenadas conocidas (x,y), tendréis que aplicar la fórmula:

(LNT(y/8)) *MOD* 4

Como cada zona de la pantalla tiene 192 cuadros numerados del 0 al 191, si queréis saber el número del cuadro donde se halla vuestro *pixel* (x,y), tendréis que aplicar:

INT(y/32) x 32 + *INT*(x/8)

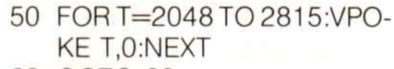
Por ejemplo, el *pixel* (203, 109) pertenece al cuadro 121 (*INT*(109/32) x 32 + *INT*(203/8) = 121) de la zona 1 de pantalla ((*INT*(109/8)*MOD*4 = 1). Si ahora llamáis «z» a la zona de pantalla y «q» al cuadro donde está vuestro *pixel* (por ejemplo, en el caso del *pixel* (203,109) sería z=1, q=121), el octeto que hay que modificar para ponerlo en función del otro sería:

Primer octeto de la tabla + *INT*(q/32) x 128 + z x 32 + q *MOD* 32 (por ejemplo, para el *pixel* (203, 109) sería el octeto 2489 (2048 + *INT*(121/32) x 128 + 1 x 32 + 121 *MOD* 32 = 2489).

La tabla 15 en la práctica

Antes de comenzar con las aplicaciones será conveniente que observéis claramente y aprendáis a distinguir las cuatro zonas de la pantalla. Para eso es este programa:

```
10 SCREEN 3:LINE (0,0) —
   (7,7),15,BF
20 LINE (0,8) — (7,15),8,BF
30 LINE (0,16) — (7,23),11,BF
40 LINE (0,24) — (7,31),4,BF
```



```
50 FOR T=2048 TO 2815:VPO-
   KE T,0:NEXT
60 GOTO 60
```

Este programa impregna de un color concreto toda una zona. Así,



la zona 0 se impregna de blanco, la zona 1 de rojo, la zona 2 de amarillo y la zona 3 de azul. En las líneas 10, 20, 30 y 40 se pintan 4 cuadros impregnados de los cuatro colores que dijimos anteriormente en el cuadro número 0 de cada una de las cuatro zonas. De esa manera, al introducir 0 en todos los octetos de la tabla 15, todos los cuadros de cada zona se ponen en función del cuadro número 0 de su propia zona, consiguiendo así que todos los cuadros de una zona adopten el mismo color y la zona sea fácil de distinguir en la pantalla.

Quizá este ejemplo en el cual



todos los cuadros adoptan el color impregnado en el cuadro 0 sea demasiado simple, pues también se pueden hacer grecas, adoptando alternativamente dos patrones,



como es el caso del programa siguiente:

```
10 SCREEN 3:PSET (0,0):PSET
   (7,7):PSET(8,7):PSET(15,0)
20 FOR T=2050 TO 2079:
   VPOKE T,T MOD 2:NEXT
30 GOTO 30
```

El patrón del cuadro 0 lo adoptan los cuadros pares y el del cuadro 1 los impares. Ambos son patrones simétricos, y gracias a eso su combinación alternativa crea esa greca que estábamos buscando. El patrón del cuadro 0 se establece con los dos primeros *PSET* de la línea 10, que consiguen que de los cuatro bloques del cuadro 0, se impregnen del color de tinta que haya en ese momento (el color de tinta que acabe de ser utilizando en un modo anterior) el bloque de arriba a la derecha y el bloque de abajo a la izquierda. Así queda este patrón exactamente igual que el carácter 193 de la tabla de caracteres MSX. (Mirad en las instrucciones de vuestro ordenador MSX.) Análogamente, el patrón del cuadro 1 queda igual que el carácter 199, es decir, el carácter simétrico al carácter 193.

Esta greca puede ser mejorada impregnando de más colores los patrones 0 y 1. (Aclaración: si el bucle *FOR-NEXT* de variable *T* de la línea 20 comienza con el octeto 2050 en vez de comenzar con el 2048 es porque los octetos 2048 y 2049 corresponden a los cuadros 0 y 1, y estos están inicialmente en función de si mismos.)

Pero aún hay más aplicaciones de la tabla 15. Imaginaros que estáis programando un juego (esquemático, claro) y uno de los elementos de vuestro juego tiene que mostrar síntomas de que está nervioso, o de que está en peligro. Lo ideal sería que se moviera muy rápidamente, como si estuviera dando saltos. Esta sería la subrutina para este caso:

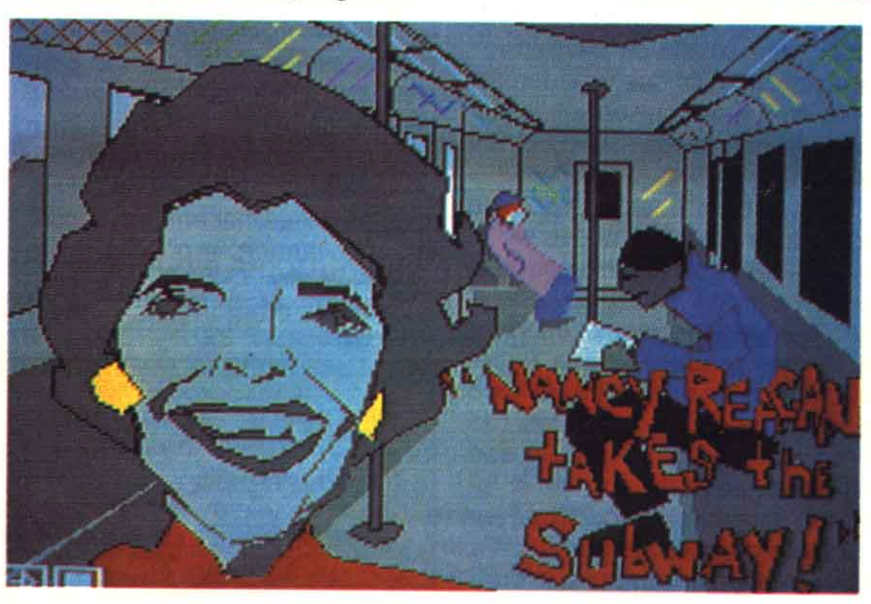
```
10 SCREEN 3:PSET (0,0)
20 FOR T=1 TO 100:VPOKE
   2061,1
30 FOR R=1 TO 80:NEXT
40 VPOKE 2060,1:VPOKE
   2061,0:FOR R=1 TO 80:
   NEXT:NEXT
```

El patrón del cuadro 0 (un bloque pintado) y el del cuadro 1 (un cuadro vacío) se alternan en dos cuadros en la parte superior de la pantalla. Eso crea un efecto de movimiento muy útil en cualquier programa de las características que antes exponíamos.

Hasta aquí las aplicaciones de las tablas del sistema de almacenamiento del modo *SCREEN 3*, el último de los cuatro modos de pantalla.

En el próximo número trataremos las figuras móviles desde el punto de vista de la memoria de vídeo, con programas y aplicaciones para el programador.

José M. Cavanillas





Definir caracteres

En el número 3 de nuestra revista, veíamos cómo se puede cambiar el juego de caracteres mediante el uso de la VRAM. Uno de los principales inconvenientes de este sistema es que al cambiar el modo de pantalla se pierden los caracteres definidos y además, estos caracteres no son válidos en los modos gráficos (SCREEN 2 y SCREEN 3).

La solución a este problema consiste en definir el nuevo juego de caracteres en algún lugar de la memoria RAM protegido por el BASIC y posteriormente, informar al sistema operativo de la dirección donde se encuentra definido el nuevo juego de caracteres. Las direcciones 63776 y 63777, son las encargadas de apuntar el lugar de la memoria donde se encuentran definidos los caracteres, por lo que modificando estas posiciones podremos alterar la dirección y dirigirla al lugar donde tenemos nuestro juego de caracteres.

Efecto de explosión

Un truco interesante, que permite hacer invisibles nuestros listados a las miradas indiscretas, consiste en pokear con el valor 255 las direcciones 32771 y 32772. En ellas se almacena el número de la primera línea del BASIC y mediante estos pokes, hacemos que contenga un número imposible, por lo que el intérprete BASIC no lo reconocerá. El programa funcionará correctamente siempre que no se encuentre con alguna instrucción GOTO. Para volver a la normalidad teclearemos los comandos siguientes:

POKE 32771,0: POKE 32772,1

Utilidades del editor de pantalla

En los manuales que acompañan los ordenadores MSX, suele venir un apéndice donde se explica las posibilidades del editor de pantalla, pero en distintas ocasiones, hemos podido comprobar que esto no es así en todos los casos. Esto nos mueve a explicar algunos de los más interesantes. Los controles ya conocidos y que tienen todos los ordenadores son; TAB, BS y INS, cuyas funciones son conocidas. Sin embargo, existen otros que no lo son tanto. Estos son:

CTRL + E: borra una línea completa de la pantalla.

CTRL + C: detiene el comando AUTO y el programa dentro de una instrucción INPUT.

CTRL + F: sitúa el cursor al principio de la palabra siguiente.

CTRL + B: sitúa el cursor en la palabra anterior.

Listados invisibles

Esta corta rutina que podéis incluir dentro de vuestros programas, produce el efecto sonoro de una explosión, a la vez que cambia el borde de la pantalla.

```
10 ON INTERVAL=1GOSUB 40:INTERVAL ON
20 SOUND 7.&B10110111:SOUND 8.31:SOUND
6,200:SOUND 11.85:SOUND 12.104:SOUND 13.0
30 FOR A=1 TO 500:NEXT:INTERVAL OFF:END
40 COLOR,,INT(RND(1)*15):RETURN
```

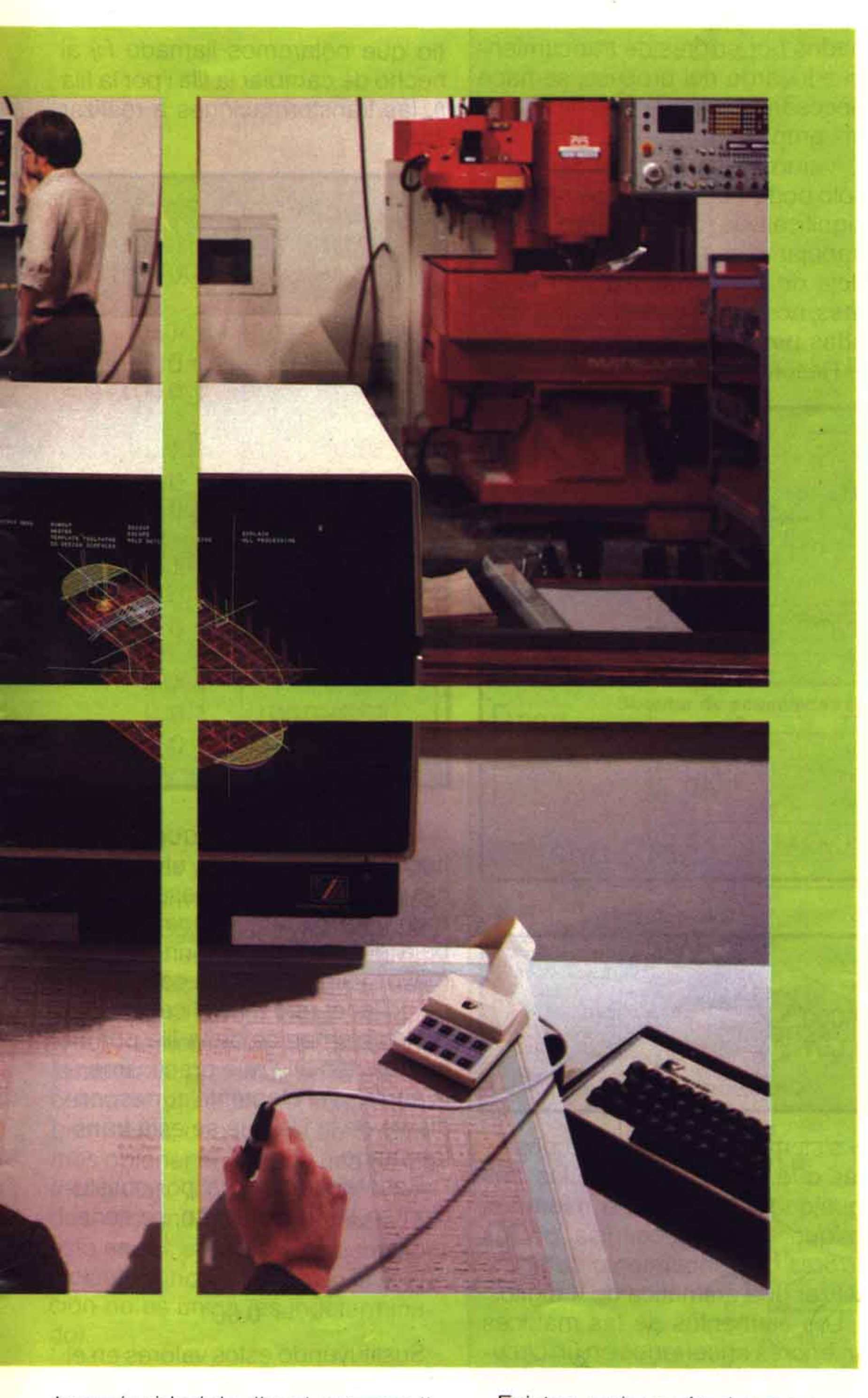


Sistemas de ecuaciones lineales

El mes pasado ya explicábamos como el ordenador ayuda a efectuar complejos cálculos matriciales, sin embargo, en esta ocasión veremos como se resuelven los sistemas de ecuaciones lineales con la imprescindible colaboración del ordenador.

En muchos aspectos, no sólo de la matemática sino también de la física y de la técnica, nos encontramos con el problema de tener que resolver un sistema de ecuaciones lineales. Sabemos resolver sin demasiado engorro un sistema que tenga 3 ó 4 ecuaciones con 3 ó 4 incógnitas, pero hay aplicaciones, como en el caso del Cálculo Matricial de Estructuras, en que nos podemos encontrar con un sistema de varias decenas de ecuaciones con varias decenas de incógnitas. Esto, evidentemente, no puede resolverse con lápiz y papel, dado el tiempo que sería necesario para ello. Sin embargo, la aparición del ordenador hizo posible la resolución de tales sistemas con un número prácticamente ilimitado de incógnitas.

El uso de un ordenador MSX permite resolver tales sistemas, pero si no utilizamos memoria ex-



terna (unidad de disco) estamos limitados por la memoria interna del ordenador, y podremos resolver así sistemas de alrededor de 50 ecuaciones con igual número de incógnitas.

Existen varios métodos para resolver sistemas de ecuaciones con ordenador. Nosotros vamos a conocer 4, y hablaremos de sus ventajas e inconvenientes. Estos 4 métodos son:

1. Método de GAUSS-JORDAN.
2. Método de JACOBI.
3. Método de iteración de GAUSS-SEIDEL.
4. Método de SOBREITERACION.

De estos 4 métodos, los tres últimos son iterativos.

El método de Gauss-Jordan

Al hablar de MATRICES Y DETERMINANTES hemos visto cómo encontrar la matriz inversa de una dada por el método de GAUSS. Para ello, lo que se hace en realidad es resolver simultáneamente 3 sistemas de ecuaciones lineales.

Vamos a conocer el método de GAUSS-JORDAN, o simplemente Método de GAUSS, resolviendo el siguiente sistema:

$$\begin{aligned} 0.001\,x + 0.001\,y + z &= 0.001 \\ x + 0.001\,y + 0.001\,z &= 0.001 \\ 0.001\,x + y + 0.001\,z &= 0.001 \end{aligned}$$

Que escribiremos en forma matricial (ver PRODUCTO DE MATRICES, en el artículo del número anterior) de la siguiente manera:

$$\begin{bmatrix} 0.001 & 0.001 & 1 \\ 1 & 0.001 & 0.001 \\ 0.001 & 1 & 0.001 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 0.001 \\ 0.001 \\ 0.001 \end{bmatrix}$$

es decir:

$$A * X = B$$

La matriz A se llama MATRIZ DE LOS COEFICIENTES y la matriz (A/B):

$$(A\ B) = \left[\begin{array}{ccc|c} 0.001 & 0.001 & 1 & 0.001 \\ 1 & 0.001 & 0.001 & 0.001 \\ 0.001 & 1 & 0.001 & 0.001 \end{array}\right]$$

se llama MATRIZ AMPLIADA.

Para resolver el sistema, bastará hacer todos los elementos por debajo de la diagonal principal (por debajo de la línea de puntos) de la matriz ampliada iguales a cero, de la forma que ya vimos en el caso de la inversa de una matriz, lo que

equivale a reducir el sistema original a otro más cómodo mediante transformaciones lineales. (Para mayor información véase «CALCULUS», Vol. 2 de Tom. M. APOSTOL.)

Resolveremos el sistema mediante transformaciones elementales, de la misma forma que vimos en el caso de la inversión de matrices, y veremos cómo, para evitar graves errores finales, originados por errores de truncamiento a lo largo del proceso, se hace necesario permutar filas de la matriz ampliada.

Vamos a suponer, primero, que sólo podemos trabajar con 4 cifras significativas (el ordenador puede trabajar con 14 cifras, lo que no deja de ser una limitación). Además, no vamos a efectuar las referidas permutaciones de filas.

Resolvamos el sistema...

$$F21(-1000) : \left[\begin{array}{ccc|c} 0.001 & 0.001 & 1 & 0.001 \\ 0 & -1 & -1.000 & -1 \\ 0.001 & 1 & 0.001 & 0.001 \end{array}\right]$$

$$F31(-1) : \left[\begin{array}{ccc|c} 0.001 & 0.001 & 1 & 0.001 \\ 0 & -1 & -1.000 & -1 \\ 0 & 1 & -1 & 0 \end{array}\right]$$

$$F32(1) : \left[\begin{array}{ccc|c} 0.001 & 0.001 & 1 & 0.001 \\ 0 & -1 & -1.000 & -1 \\ 0 & 0 & -1.001 & -1 \end{array}\right]$$

Nuestro sistema es, ahora:

$$\begin{aligned} 0.001\,x + 0.001\,y + \quad z &= 0.001 \\ -\,1.000\,z &= -1 \\ -\,1.001\,z &= -1 \end{aligned}$$

que se puede resolver de abajo hacia arriba. De la tercera ecuación:

$$z = \frac{1}{1001} = 0.001$$

Sustituyendo en la segunda ecuación:

$$y = \frac{-1 + 1.000 \times 0.001}{0.001} = 0$$

y sustituyendo los valores de **y** y **z** en la primera ecuación:

$$x = \frac{0.001 - 0.001}{0.001} = 0$$

Si se sustituyen estos valores en el sistema inicial, se podrá observar que no se cumplen las tres igualdades miembro a miembro, lo que es consecuencia de los errores de truncamiento debido a utilizar una aritmética de 4 dígitos.

Los elementos de las matrices anteriores encerrados en un círculo son los que se han utilizado para anular los elementos por debajo de la diagonal principal, y son conocidos como PIVOTES.

Si, en lugar de resolver el sistema como lo hemos hecho, lo hacemos permutando algunas filas (lo que notaremos llamado *Fij* al hecho de cambiar la fila i por la fila j), las transformaciones a realizar son:

$$F21 : \left[\begin{matrix} 1 \\ 0.001 \\ 0.001 \end{matrix}\right.$$

$$F21(-0.001) : \left[\begin{matrix} 1 \\ 0 \\ 0.001 \end{matrix}\right.$$

$$F31(-0.001) : \left[\begin{matrix} 1 \\ 0 \\ 0 \end{matrix}\right.$$

$$F32 : \left[\begin{matrix} 1 \\ 0 \\ 0 \end{matrix}\right.$$

$$F32(-0.001) : \left[\begin{matrix} 1 \\ 0 \\ 0 \end{matrix}\right.$$

Obsérvese que lo que hemos hecho ahora ha sido utilizar como pivote, en cada caso, el elemento de cada columna, por debajo de la diagonal principal, de mayor valor absoluto, con objeto de no tener que multiplicar el resto de los elementos de la fila por un número tan alto que prácticamente anularía al elemento correspondiente de la fila que se está transformando.

Resolviendo ahora por sustitución en sentido inverso, se tiene:

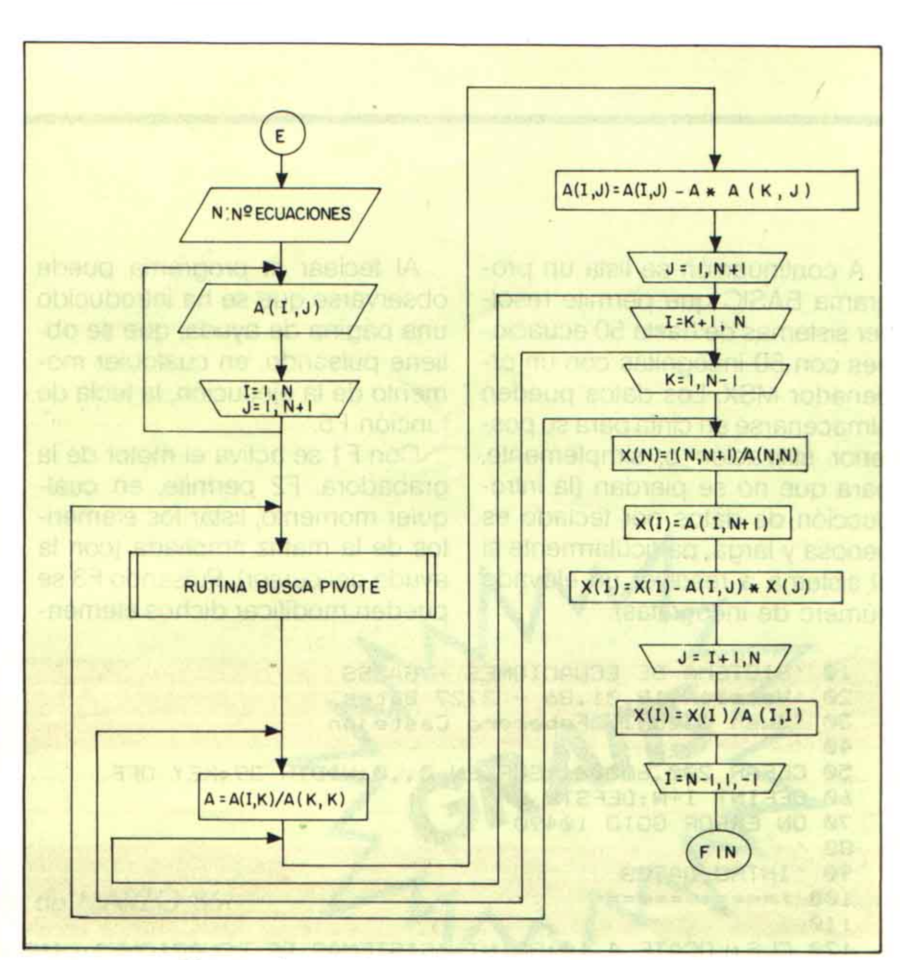
$$\begin{aligned} z &= 0.001 \\ y &= 0.001 \\ x &= 0.001 \end{aligned}$$

Sustituyendo estos valores en el sistema original, vemos que estamos más próximos a la verdadera solución que en el caso anterior. No obstante, sigue habiendo un pequeño error, debido a la acumulación de los errores de truncamiento.

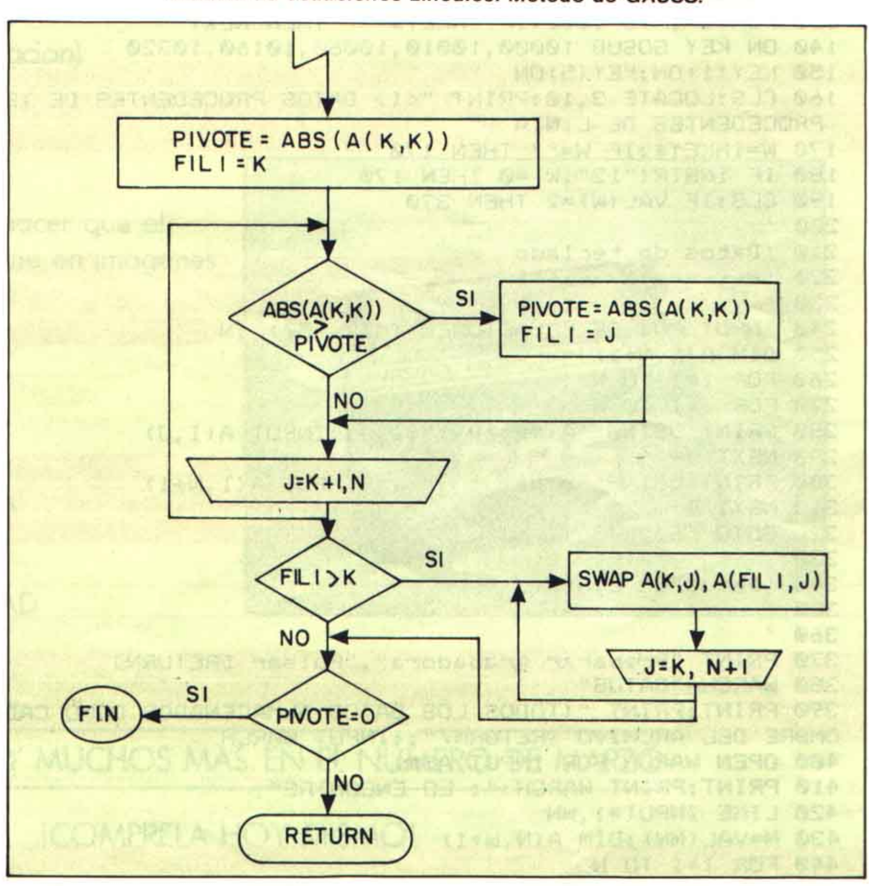
0.001	0.001	0.001
0.001	1	0.001
	0.001	0.001
0.001	0.001	0.001
0.001	1	0.001
1	0.001	0.001
0.001	0.001	0.001
0.001	1	0.001
1	0.001	0.001
0.001	0.001	0.001
1	0.001	0.001
0.001	1	0.001
0.001	0.001	0.001
1	0.001	0.001
0	1	0.001

El hecho de realizar intercambios de filas y utilizar el método de las transformaciones elementales o de GAUSS, nos permite, además, conocer si estamos ante un sistema sin solución o con más de una, ya que si el pivote fuera un cero, todos los elementos por debajo de la diagonal principal, en la misma columna que el pivote, serían también ceros, con lo que no podríamos obtener el valor de una de las incógnitas, ya que al sustituir tendríamos una división por cero. Si esto es así, el sistema no tiene solución (es incompatible), o la solución no es única (es indeterminado).

Los diagramas de flujo de las figuras 1 y 2 sintetizan el proceso completo del método que estamos estudiando. Para una mayor claridad se ha desglosado (diagrama de la figura 2) la rutina de búsqueda del pivote.



Sistema de ecuaciones Lineales: Método de GAUSS.



Método de GAUSS. Rutina busca pivote.

A continuación se lista un programa BASIC que permite resolver sistemas de hasta 50 ecuaciones con 50 incógnitas con un ordenador MSX. Los datos pueden almacenarse en cinta para su posterior utilización o, simplemente, para que no se pierdan (la introducción de datos por teclado es penosa y larga, particularmente si el sistema a resolver un elevado número de incógnitas).

Al teclear el programa puede observarse que se ha introducido una página de ayuda, que se obtiene pulsando, en cualquier momento de la ejecución, la tecla de función F5.

Con F1 se activa el motor de la grabadora. F2 permite, en cualquier momento, listar los elementos de la matriz ampliada (con la ayuda del cursor). Pulsando F3 se pueden modificar dichos elementos y con F4 los datos pueden ser grabados en cinta.

Una vez obtenida en pantalla la solución al sistema, se puede listar la matriz resultante tras el proceso de transformación. Desde aquí (y siempre desde el listado) se puede volver a realizar el proceso pulsando de nuevo la barra espaciadora.

J. Antonio Feberero

```
10 'SISTEMA DE ECUACIONES - GAUSS
20 'Versión 19.01.86 - 3727 Bytes
30 'Juan Antonio Feberero Castejón
40 '
50 CLEAR 200,60000!:SCREEN 0,,0:WIDTH 39:KEY OFF
60 DEFINT I-N:DEFSTR W
70 ON ERROR GOTO 10490
80 '
90 'INTRO DATOS
100 '===========
110 '
120 CLS:LOCATE 4,10:PRINT "SISTEMAS DE ECUACIONES LINEALES",,,TAB(12);"METODO DE
 GAUSS
130 FOR I=1 TO 1000:IF INKEY$="" THEN NEXT
140 ON KEY GOSUB 10000,10010,10060,10160,10320
150 KEY(1)ON:KEY(5)ON
160 CLS:LOCATE 3,10:PRINT "<1> DATOS PROCEDENTES DE TECLADO",,,TAB(3);"<2> DATOS
 PROCEDENTES DE CINTA
170 W=INKEY$:IF W="" THEN 170
180 IF INSTR("12",W)=0 THEN 170
190 CLS:IF VAL(W)=2 THEN 370
200 '
210 'Datos de teclado
220 '----------------
230 '
240 INPUT "Nº DE ECUACIONES (Máx. 50)";N
250 DIM A(N,N+1)
260 FOR I=1 TO N
270 FOR J=1 TO N
280 PRINT USING "A(##;##):";I;J;:INPUT A(I,J)
290 NEXT J
300 PRINT USING "B(##)    :";I;:INPUT A(I,N+1)
310 NEXT I
320 GOTO 560
330 '
340 'Datos de cinta
350 '--------------
360 '
370 PRINT "Preparar grabadora","Pulsar [RETURN]
380 WARCH="DATOS"
390 PRINT:PRINT "(TODOS LOS DATOS ALMACENADOS COMO CADE-NAS DE CARACTERES)",,,"N
OMBRE DEL ARCHIVO (RETURN)";:INPUT WARCH
400 OPEN WARCH FOR INPUT AS#1
410 PRINT:PRINT WARCH;": LO ENCONTRE"
420 LINE INPUT#1,WN
430 N=VAL(WN):DIM A(N,N+1)
440 FOR I=1 TO N
450 FOR J=1 TO N+1
```









¡ATENCION! Con el ejemplar de MARZO 86

PC MAGAZINE le regala un DISKETTE DEMO DE PC-TEXT-PC-BASE.
(Obsequio de Quinta Generación)

Además en este número:

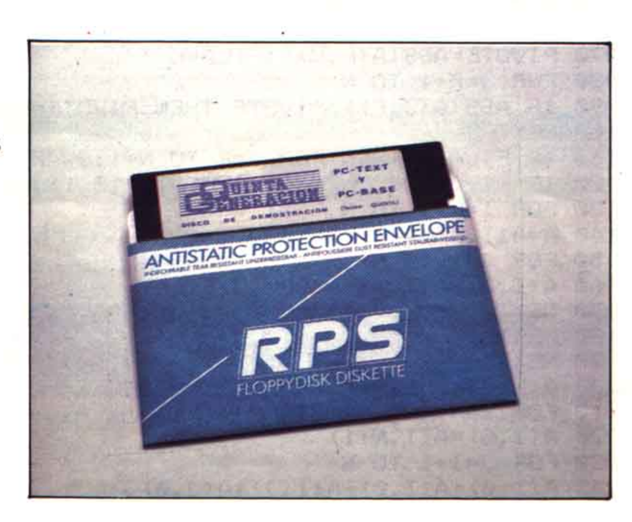
GRAFICOS:
Hardware y Software para hacer que el ordenador personal transforme en imágenes sus ideas y cifras, etc.

Compatible

SPERRY PC/HT

Software
ENCORE! HOJA FINANCIERA

Programación
VTREE, UN ARBOL DE VERDAD



ESTOS TEMAS Y MUCHOS MAS EN EL NUMERO DE MARZO

¡COMPRELA HOY MISMO!

```
460 IF EOF(1) THEN 510
470 LINE INPUT#1,W
480 A(I,J)=VAL(W)
490 NEXT J
500 NEXT I
510 CLOSE #1
520 '
530 'LIST. DATOS
540 '===========
550 '
560 CLS
570 KEY(2)ON:KEY(3)ON:KEY(4)ON
580 FOR I=1 TO N
590 FOR J=1 TO N+1
600 IF J=N+1 THEN PRINT USING "B(##)    =&";I;STR$(A(I,N+1)):GOTO 640
610 PRINT USING "A(##;##)=&";I;J;STR$(A(I,J))
620 IF CSRLIN=20 THEN 640
630 GOTO 700
640 W=INKEY$:IF W="" THEN 640
650 IF W=CHR$(30) AND I*J<N*N+N THEN 690
660 IF W=CHR$(31) AND J<N+2 THEN J=J-CSRLIN-20:IF J<-1 THEN I=I-2:GOTO 710 ELSE
690
670 IF W=CHR$(32) THEN 5040
680 GOTO 640
690 CLS
700 NEXT J
710 CLS:IF I<0 THEN I=0
720 NEXT I
5000 '
5010 'PROCESO
5020 '=======
5030 '
5040 CLS:LOCATE 15,11:PRINT "CALCULANDO...
5050 KEY(3)OFF:KEY(4)OFF
5060 FOR K=1 TO N-1
5070 PIVOTE=ABS(A(K,K)):FILA=K
5080 FOR J=K+1 TO N
5090 IF ABS(A(J,K))>PIVOTE THEN PIVOTE=ABS(A(J,K)):FILA=J
5100 NEXT J
5110 IF FILA>K THEN FOR J=K TO N+1:SWAP A(K,J),A(FILA,J):NEXT
5120 IF PIVOTE=0 THEN CLS:LOCATE 10,11:PRINT "SISTEMA NO RESOLUBLE":END
5130 FOR I=K+1 TO N
5140 A=A(I,K)/A(K,K)
5150 FOR J=1 TO N+1
5160 A(I,J)=A(I,J)-A*A(K,J)
5170 NEXT J
5180 NEXT I
5190 NEXT K
5200 A(N,0)=A(N,N+1)/A(N,N)
5210 FOR I=N-1 TO 1 STEP-1
5220 A(I,0)=A(I,N+1)
5230 FOR J=I+1 TO N
5240 A(I,0)=A(I,0)-A(I,J)*A(J,0)
5250 NEXT J
5260 A(I,0)=A(I,0)/A(I,I)
5270 NEXT I
6000 '
6010 'Resultados
6020 '----------
6030 '
6040 CLS
6050 FOR I=1 TO N
6060 PRINT USING "x(##)=&";I;STR$(A(I,0))
6070 IF CSRLIN=20 OR I=N THEN 6090
```

```
6080 GOTO 6150
6090 W=INKEY$:IF W="" THEN 6090
6100 IF W=CHR$(30) AND I<N THEN 6140
6110 IF W=CHR$(31) AND I>20 THEN I=I-CSRLIN-20:GOTO 6140
6120 IF W=CHR$(32) THEN 560
6130 GOTO 6090
6140 CLS
6150 NEXT I
10000 MOTOR:RETURN
10010 RETURN 560
10020 '
10030 'Modifica
10040 '--------
10050 '
10060 CLS
10070 INPUT "Dato a modificar: FILA";I
10080 INPUT "COLUMNA (0 para término independ.";J
10090 IF J=0 THEN J=N+1
10100 INPUT "NUEVO VALOR";A(I,J)
10110 RETURN 560
10120 '
10130 'GRABA CINTA
10140 '===========
10150 '
10160 CLS:PRINT "Preparar grabadora","Pulsar [RETURN]
10170 WARCH="DATOS"
10180 PRINT:PRINT "(TODOS LOS DATOS SE ALMACENAN COMO CA- DENAS DE CARACTERES)",
,,"NOMBRE DEL ARCHIVO (RETURN)";:INPUT WARCH
10190 OPEN WARCH FOR OUTPUT AS#1
10200 PRINT#1,STR$(N)
10210 FOR I=1 TO N
10220 FOR J=1 TO N+1
10230 PRINT#1,STR$(A(I,J))
10240 NEXT J
10250 NEXT I
10260 CLOSE #1
10270 RETURN 560
10280 '
10290 'Help
10300 '----
10310 '
10320 XP=POS(0):YP=CSRLIN
10330 FOR IHELP=1 TO 960:POKE 60000!+IHELP,VPEEK(IHELP):NEXT
10340 CLS
10350 PRINT "HELP",,"────"
10360 PRINT:PRINT "F1:Motor",,"F2:List. datos","F3:Modific. datos","F4:Grabar da
tos en cinta","F5:Help
10370 PRINT:PRINT:PRINT "LIS. DATOS",,STRING$(10,195)
10380 PRINT "Cursor:Página arriba/abajo","Barra espaciadora:Sale de List. Comien
-za proceso. Vuelve a List.",,,,"En List. están activadas F1 a F5","En otro caso
 están activadas F1, F2 Y  F5
10390 LOCATE 12,23:PRINT "Pulsa una tecla";
10400 IF INKEY$="" THEN 10400
10410 CLS
10420 FOR IHELP=1 TO 960:VPOKE IHELP,PEEK(60000!+IHELP):NEXT
10430 LOCATE XP,YP
10440 RETURN
10450 '
10460 'Errores
10470 '-------
10480 '
10490 IF ERR=11 AND ERL=5200 OR ERL=5140 THEN 10510
10500 CLS:LOCATE 7,12:PRINT "ERROR ";ERR;" EN LINEA ";ERL:STOP
10510 CLS:LOCATE 9,11:PRINT "SISTEMA NO RESOLUBLE.":STOP
```

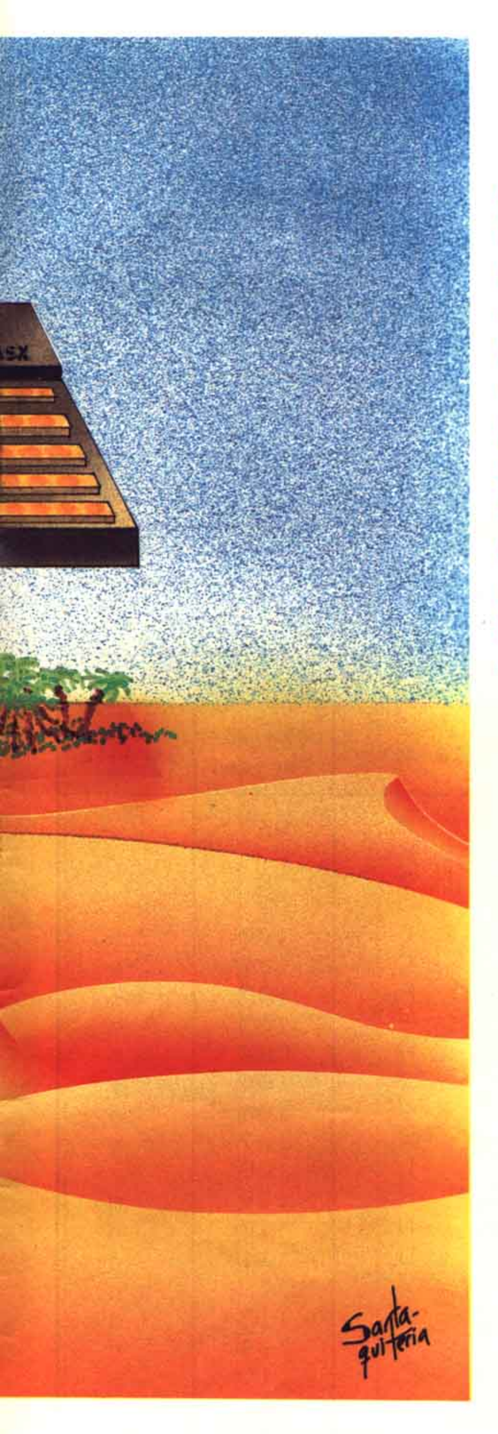


Existen varios tipos de lenguajes *BASIC* diferentes, pero, en general, todos ellos tienen funciones similares o bastante parecidas. Además aquí no nos limitaremos a enumerar las funciones y sentencias de *BASIC,* sino que daremos ejemplos de utilización que te pueden orientar sobre para qué sirven determinadas funciones y sentencias que, en principio, parecen no tener interés. Por eso, aunque tu ordenador no sea MSX, podrás encontrar cierta utilidad al curso. Más adelante publicaremos un cuadro comparativo de las distintas versiones de *BASIC.*

Por tanto trataremos de que el curso sea un curso DINAMICO, no limitándonos a describir lo que se puede hacer, sino que pondremos ejemplos de cómo hacerlo y para qué.

Bits y Bytes

Te decíamos que al conectar el ordenador, éste no hace nada; simplemente aparece algo arriba a la izquierda que, entre otras co-



Es posible que al conectar por primera vez tu ordenador te hayas llevado una decepción: el ordenador no hace nada. Probablemente hayas aprendido más o menos, mucho o poco a utilizarlo. Puede que seas un gran experto, con lo que este Curso de BASIC te servirá de poco; pero también puede que no conozcas algunas o muchas de las funciones y posibilidades del BASIC. Por eso vamos a comenzar aquí y ahora nuestro CURSO DE BASIC, orientado en principio para tu ordenador MSX, pero que, en términos generales te puede servir para otros ordenadores.

sas, dice que el ordenador está preparado para trabajar en *BASIC* y que hay 28815 *BYTES FREE* (en castellano: 28815 OCTETOS LIBRES). Si tu ordenador es de menos de 64 *Kbytes,* en lugar de 28815 aparecerá un número inferior. Esto quiere decir que, si tu ordenador es de 64 *Kbytes,* sólo dispones de 28815 *bytes* u octetos. Te preguntarás qué ha pasado con los 64 *Kbytes.* Para explicártelo vamos a comentar un poco cómo está estructurado un ordenador; y empezaremos por ver qué es un *BIT* y qué es un *BYTE:*

Imagínate que el ordenador es un fichero. Todos los datos, sentencias e información que va a manejar el ordenador deben ser almacenados o MEMORIZADOS en ese fichero. Si, por ejemplo, queremos que el ordenador escriba en la pantalla el número 10000, le diremos:

ESCRIBE EN LA PANTALLA 10000

Pues bien, lo primero, es decir: «ESCRIBE EN LA PANTALLA» es una SENTENCIA (con lo que le decimos al ordenador QUE DEBE HACER), y el ordenador lo almacenará en la primera ficha que tenga libre; y en la segunda ficha memorizará el número 10000, así:

FICHA N.° 1: «ESCRIBE EN LA PANTALLA».

FICHA N.° 2: «10000».

Cuando tú le digas al ordenador que EJECUTE o haga esa operación que le has indicado anteriormente, éste buscará la primera ficha y leerá: «ESCRIBE EN LA PANTALLA», con lo que se preparará para escribir en la pantalla lo que esté indicado en la ficha siguiente, es decir «10000». Más adelante vermos lo sencillo que resulta hacer todo esto.

Pero, evidentemente, el ordenador no tiene ningún fichero, ni, por supuesto, fichas. El ordenador tiene una MEMORIA, de la que hablaremos más adelante, en lugar de fichero; y esta memoria está compuesta de *BYTES* u *OCTETOS,* que corresponden a la fichas (ver figura 1).

Cada octeto está compuesto de 8 *BITS.* Un *BIT* es la unidad de memoria más pequeña; algo así como la célula de un ordenador.

Suponte que cada octeto está compuesto de 8 bombillas muy pequeñas (ver figura 2). Numeramos las bombillas del 0 al 7 y de derecha a izquierda, como en el dibujo. Si una bombilla está encendida es que por ella pasa corriente, y si está apagada no pasa corriente.

Si por una bombilla, por ejemplo la 4, pasa corriente, el ordenador leerá un 1, y si no pasa, un 0, de forma que:

PASA CORRIENTE = 1

NO PASA CORRIENTE = 0

Así, si está encendida sólo la bombilla 4, el ordenador leerá:

7	6	5	4	3	2	1	0
0	0	0	1	0	0	0	0

Por tanto vemos que un *bit* puede estar a 0 o a 1, y un octeto estará compuesto exclusivamente de ceros y unos.

Entonces: ¿Cómo escribir el número 7, o la letra Z mayúscula, o el signo +? Muy sencillo:

En base 10, que es la que utilizamos normalmente, trabajamos con los números cero (0), uno (1), dos (2),... nueve (9), es decir, con 10 unidades. Igualmente, en base 2 se trabaja con dos unidades: el cero (0) y el uno (1).

No vamos a ver ahora cómo se pasa un número de la base 10 a la base 2, pero si vamos a ver cómo se pasa un número de ocho cifras en la base 2 al número correspondiente en la base 10. Para ello hay que saber que:

$2^0 = 1$

$2^1 = 2$

$2^2 = 2*2 = 4$

$2^3 = 2*2*2 = 8$

$2^4 = 2*2*2*2 = 16$

$2^5 = 2*2*2*2*2 = 32$

$2^6 = 2*2*2*2*2*2 = 64$

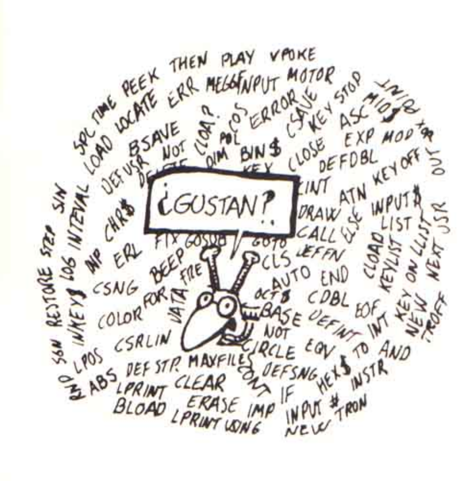
$2^7 = 2*2*2*2*2*2*2* = 128$

De forma que el ordenador lee el número:

7	6	5	4	3	2	1	0
A	B	C	D	E	F	G	H

donde A, B, C,..., H pueden ser o cero o uno, así:

Código Hexadecimal	00-1F		20-3F		40-5F		60-7F	
	Código Decimal	Carácter	Código Decimal	Carácter	Código Decimal	Carácter	Código Decimal	Carácter
0	0		32		64	@	96	`
1	1	☺	33	!	65	A	97	a
2	2	☻	34	"	66	B	98	b
3	3	♥	35	#	67	C	99	c
4	4	♦	36	$	68	D	100	d
5	5	♣	37	%	69	E	101	e
6	6	♠	38	&	70	F	102	f
7	7	•	39	'	71	G	103	g
8	8	◘	40	(	72	H	104	h
9	9	○	41	)	73	I	105	i
A	10	◙	42	*	74	J	106	j
B	11	♂	43	+	75	K	107	k
C	12	♀	44	,	76	L	108	l
D	13	♪	45	-	77	M	109	m
E	14	♫	46	.	78	N	110	n
F	15	☼	47	/	79	O	111	o
0	16	┼	48	0	80	P	112	p
1	17	┴	49	1	81	Q	113	q
2	18	┬	50	2	82	R	114	r
3	19	┤	51	3	83	S	115	s
4	20	├	52	4	84	T	116	t
5	21	┼	53	5	85	U	117	u
6	22	│	54	6	86	V	118	v
7	23	─	55	7	87	W	119	w
8	24	┌	56	8	88	X	120	x
9	25	┐	57	9	89	Y	121	y
A	26	└	58	:	90	Z	122	z
B	27	┘	59	;	91	[	123	{
C	28	╳	60	<	92	\	124	¦
D	29	╱	61	=	93	]	125	}
E	30	╲	62	>	94	^	126	~
F	31	+	63	?	95	_	127	



ABCDEFGH = $A*2^7 + B*2^6 + C*2^5 + D*2^4 + E*2^3 + F*2^2 + G*2^1 + H*2^0$.

El número anterior (10000) será igual (en base 10) a:

$10000_{(2)} = 1*2^4 + 0*2^3 + 0*2^2 + 0*2^1 + 0*2^0 = 1*16 + 0*8 + 0*4 + 0*2 + 0*1 = 16_{(10)}$

y el número 10111001 será:

$10111001_{(2)} = 1*2^7 + 0*2^6 + 1*2^5 + 1*2^4 + 1*2^3 + 0*2^2 + 0*2^1 + 1*2^0 = 1*128 + 0*64 + 1*32 + 1*16 + 1*8 + 0*4 + 0*2 + 1*1 = 185_{(10)}$

El valor más alto que puede tomar un octeto será, cuando todos los *bits* estén a 1:

$1\,1\,1\,1\,1\,1\,1\,1_{(2)} = 255_{(10)}$

y el más bajo, cuando todos los octetos estén a 0:

$0_{(2)} = 0_{(10)}$

Por tanto podremos poner los ceros y unos de los ocho bits de

Código Hexadecimal	00-1F		20-3F		40-5F		60-7F	
	Código Decimal	Carácter	Código Decimal	Carácter	Código Decimal	Carácter	Código Decimal	Carácter
0	128	Ç	160	á	192	▁	224	α
1	129	ü	161	í	193	▚	225	β
2	130	é	162	ó	194	▆	226	Γ
3	131	â	163	ú	195	▔	227	π
4	132	ä	164	ñ	196	▬	228	Σ
5	133	à	165	Ñ	197	▥	229	σ
6	134	å	166	ª	198	▌	230	µ
7	135	ç	167	º	199	▞	231	γ
8	136	ê	168	¿	200	▓	232	Φ
9	137	ë	169	⌐	201	▏	233	θ
A	138	è	170	¬	202	▒	234	Ω
B	139	ï	171	½	203	//	235	δ
C	140	î	172	¼	204	\\	236	∞
D	141	ì	173	¡	205	▼	237	ø
E	142	Ä	174	«	206	▲	238	ε
F	143	Å	175	»	207	▶	239	∩
0	144	É	176	Ã	208	◀	240	≡
1	145	æ	177	ã	209	⧗	241	±
2	146	Æ	178	Ĩ	210	⧓	242	≥
3	147	ô	179	ĩ	211	▪	243	≤
4	148	ö	180	Õ	212	▪	244	⌠
5	149	ò	181	õ	213	▪	245	⌡
6	150	û	182	Ũ	214	▪	246	÷
7	151	ù	183	ũ	215	※	247	≈
8	152	ÿ	184	Ĳ	216	Λ	248	°
9	153	Ö	185	ĳ	217	∓	249	∙
A	154	Ü	186	¾	218	ω	250	·
B	155	¢	187	∽	219	█	251	√
C	156	£	188	◊	220	▄	252	ⁿ
D	157	¥	189	‰	221	▌	253	²
E	158	₧	190	¶	222	▐	254	■
F	159	ƒ	191	§	223	▀	255	

un octeto de 256 formas distintas.

El código ASCII

A cada una de esas 256 formas distintas de combinar los ceros y unos de un octeto se le asigna un CARACTER o un signo. Por ejemplo al número 65 (01000001) se le asigna la letra A mayúscula, al número 53 (00110101) se le asigna el número 5, el número 7 (00000111) significa «emitir un pitido» y el número 13 (00001101) tiene el mismo significado que cuando se pulsa la tecla *RETURN* o *ENTER*.

Para que todos los ordenadores MSX sean compatibles, todas las combinaciones anteriores deben tener el mismo significado cualquiera que sea la marca del ordenador, por ejemplo, en todos los ordenadores MSX, el número 65 (01000001) se asigna a la letra A. (Esto también suele ocurrir con otros ordenadores que no son MSX.)

En el cuadro 1 podeis encontrar todos los signos que se asignan a cada una de las 256 formas de combinar los ceros y unos de un octeto. A este cuadro se le llama *Código ASCII*, y a cada uno de los números que identifican a cada carácter, se les llama simplemente códigos. Los códigos 0 al 32 son códigos de control y los iremos conociendo a medida que avanza el curso. El código 255 es una copia del cursor.

Por ahora no encontraremos ninguna utilidad al código *ASCII* y sólo nos sirve para saber como interpreta el ordenador los caracteres con los que puede trabajar. Más adelante veremos lo útil que es conocer dicho código en muchos casos.

ROM, RAM, VRAM

Cualquier ordenador está formado por:

- La UNIDAD CENTRAL DE PROCESO *(U.C.P.)*
- La UNIDAD ARITMETICO-LOGICA *(U.A.L.)*
- La MEMORIA PRINCIPAL *(M.P.)*
- Los PERIFERICOS

Podemos representar esto en un sencillo dibujo esquemático (ver figura 3). (Las flechas indican el sentido en que se transfiere la información.)

La UNIDAD CENTRAL DE PROCESOO *(U.C.P.)* es la encargada de recibir toda la información, trabajar con ella si es preciso, y enviarla a donde corresponda. La información que llega a la *U.C.P.* puede proceder de la memoria principal, de la unidad aritmético-lógica o de un periférico (cinta magnetofónica, unidad de disco o teclado). Una vez PROCESADA la información, ésta puede ir destinada a la *M.P.*, la *U.A.L.* o un periférico (cinta magnetofónica, unidad de disco, monitor, o impresora).

La UNIDAD ARITMETICO-LOGICA *(U.A.L.)* se encarga de realizar las operaciones aritméticas (suma, resta, multiplicación...) y las lógicas (comparar dos números o

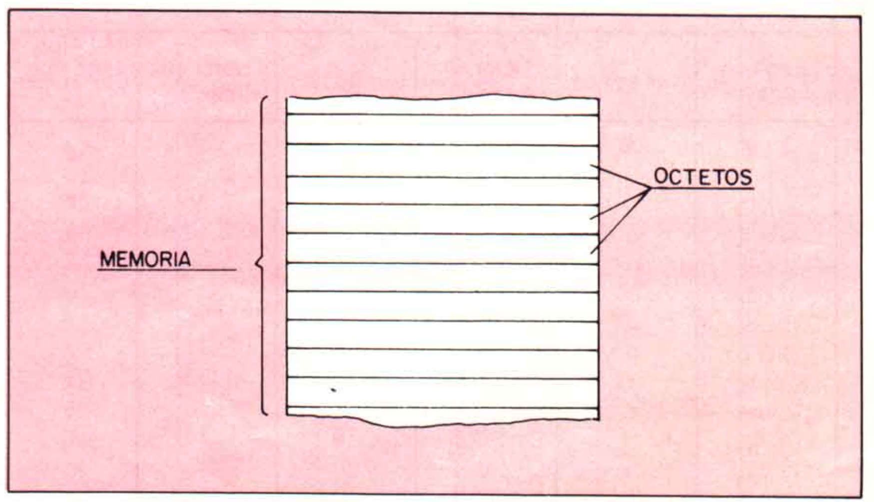


Figura 1



expresiones,...). Los datos que llegan a la *U.A.L.* proceden de la *U.C.P.* y el resultado vuelve a la *U.C.P.*, que es la que tiene el control.

La MEMORIA PRINCIPAL *(M.P.)* sirve para almacenar o memorizar la información que se está tratando.

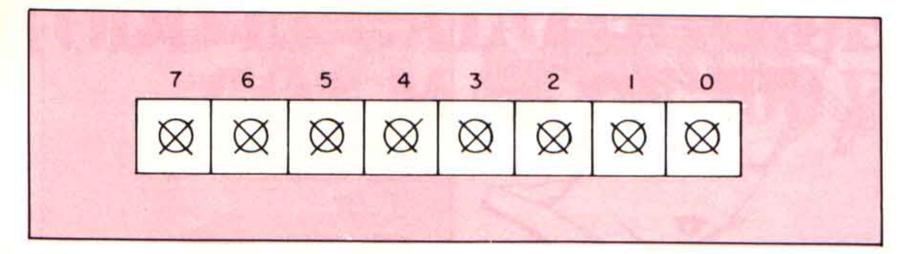


Figura 2

Los PERIFERICOS constituyen la comunicación del ordenador con el exterior. Sirven para: almacenar información, como auxilio de la memoria (cinta magnetofónica, unidad de disco); enviar información al ordenador (cinta magnetofónica, unidad de disco, teclado); o recibir información del ordenador (cinta magnetofónica, unidad de disco, monitor, impresora).

Podemos considerar la memoria del ordenador MSX como dividida en tres partes: la *ROM*, la *RAM* y la *VRAM*.

La *ROM (READ ONLY MEMORY)*, es la MEMORIA DE SOLO LECTURA. El ordenador puede solamente LEER cosas en esa memoria y no puede ESCRIBIR nada en ella. En esta memoria están almacenadas las instrucciones que le indican al ordenador lo que debe hacer en cada caso (cuando queremos que sume dos números o que escriba algo en la impresora, por ejemplo). Los ordenadores MSX tienen 32 *Kbytes* de memoria *ROM*. (Ver nota al pie.)

NOTA: 1 *Kbyte* = 2^{10} *bytes* = 1024 *bytes*. Así 32 *Kbytes* = 32*1024 = 32768 *bytes*.

La *RAM (RANDOM ACCESS MEMORY)* es la MEMORIA DE ACCESO ALEATORIO. En ella el ordenador puede tanto leer como escribir. Así, por ejemplo, cuando escribimos un programa en *BASIC* éste se almacena en *RAM*.

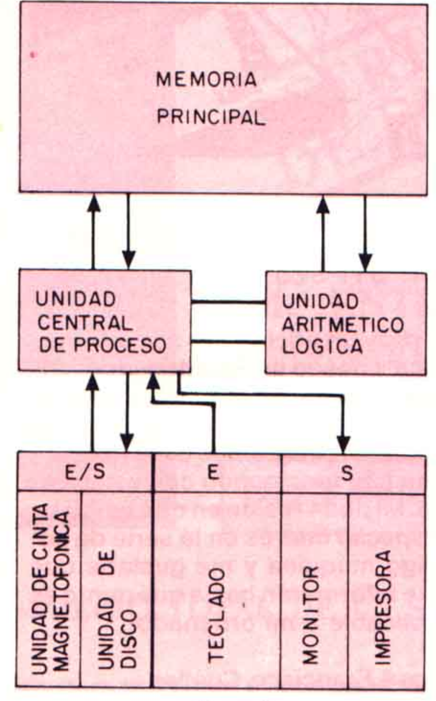


Figura 3

Cuando damos a una variable (A) un valor 31 (A = 31), el ordenador almacena el nombre de la variable y su valor en algún lugar de la *RAM*.

La *RAM* de un ordenador MSX de 64 *Kbytes* tiene una capacidad de 64 *Kbytes*.

La *VRAM (VIDEO RANDOM ACCESS MEMORY)* es la MEMORIA DE VIDEO DE ACCESO ALEATORIO. Es parecida a la *RAM* (podemos leer y escribir en ella) pero sirve exclusivamente para controlar la pantalla del monitor. Con la *VRAM* podemos determinar cuál va a ser el color de fondo de la pantalla, o hacer que aparezca la letra A mayúscula en la parte superior izquierda de la pantalla.

La *VRAM* tiene una capacidad de 16 *Kbytes* en un ordenador MSX.

El ordenador trabaja en su lenguaje particular de ceros y unos. Por tanto, si nosotros queremos que nos entienda cuando escribimos un programa en *BASIC*, deberá poder INTERPRETAR lo que hemos escrito. Para ello, es decir para TRADUCIR el programa en *BASIC* (que es lo que se llama un Lenguaje de Alto Nivel, es decir un lenguaje que el ordenador debe traducir para poder entenderlo), el ordenador necesita un INTERPRETE. Pues bien, el ordenador tiene almacenado en su memoria *ROM* un INTERPRETE o TRADUCTOR para leer el programa en *BASIC* y traducirlo a su lenguaje de ceros y unos. Cuando conectamos el ordenador, éste, automáticamente, lee todo lo que está escrito en *ROM* y lo escribe en *RAM*, para poder trabajar. Como consecuencia, nuestra memoria *RAM* se ve reducida considerablemente, hasta esos 28815 *bytes* que nos quedan para poder utilizar. Como la *RAM* tiene 64 *Kbytes* y la *ROM* tiene 32 *Kbytes*, es de esperar que nos queden libres 32 *Kbytes* y no los aproximadamente 28 *Kbytes* que en realidad nos quedan. En realidad, esa diferencia de aproximadamente 4 *Kbytes* se utilizan por el ordenador para permitir que nos podamos comunicar con él desde el exterior, es decir en las vías de acceso o de entradas y salidas *(E/S)*.

Rincón del lector

COMPATIBILIDAD ENTRE LA PRIMERA Y SEGUNDA GENERACION DE MSX

Tengo un ordenador Sony HB-75P, con unidad de disco e impresora y tengo la siguiente duda. ¿Serán compatibles los ordenadores MSX de la segunda generación con los que existen actualmente?

José Manuel García
Málaga

Según las noticias que poseemos, tanto los ordenadores como los periféricos existentes actualmente en el mercado son compatibles con la segunda generación MSX, al fin y al cabo, la idea de MSX es mantener la compatibilidad entre todos los elementos que compongan el estándar.

CONECTOR DE CASSETTE

Poseo un Spectravideo 728 y un cassette. El problema es que cuando compré el ordenador, no traía conexión de cassette, he buscado en muchas tiendas especializadas en electrónica y no he podido conseguirlo. Les ruego que me indiquen si puedo usar un cable de otro MSX sin dañar mi ordenador.

José Jorge Morales
Granada

Resulta extraño que al comprar un ordenador de este tipo no le incluyan el cable del cassette, cuando TODOS los fabricantes de MSX lo dan con el resto del equipo. Debe exigir de dicho distribuidor que le entreguen el cable correspondiente, ya que esta negativa por parte de ellos traería consecuencias negativas, tanto para la imagen de Spectravídeo como para la empresa que lo distribuye. De cualquier modo, la conexión del casette es igual en todos los ordenadores del sistema MSX, por lo tanto, podrá utilizar el cable de otro ordenador del estándar.



CODIGO MAQUINA PARA EL SVI-328

Soy un aficionado a la informática y poseo un Spectravideo 328, que aunque no es MSX es bastante similar. De hecho he probado algunos programas de la revista y me han funcionado correctamente. Mi duda reside en que tengo un especial interés en la serie de código máquina y me gustaría que me informaran hasta qué punto es aplicable a mi ordenador.

José Francisco Cuéllar
Alicante

Todos los conceptos explicados en la serie de código máquina son aplicables a todos los ordenadores que utilicen el microprocesador Z-80. Las diferencias radican en que las llamadas a subrutinas de la ROM varían del sistema MSX al Spectravideo 328, por lo que, un programa que haga referencia a rutinas no funcionará correctamente en este último.

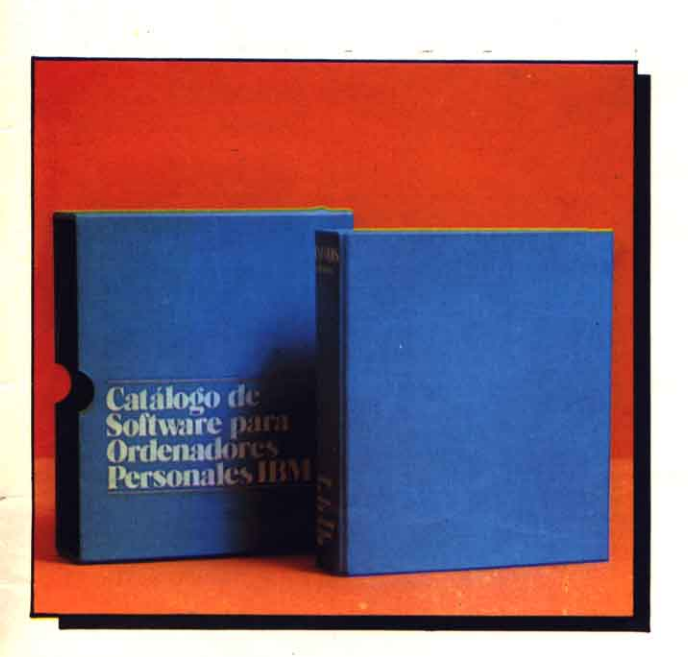
DIRECTOR:
Juan Arencibia.
COORDINADOR EDITORIAL:
J. Ignacio Rey.
COLABORADORES:
Octavio López, Angel Zarazaga, Teresa Aranda, Ricardo García.
DISEÑO:
Benito Gil y Ricardo Segura.
Editada por:
PUBLINFORMATICA, S.A.
PRESIDENTE:
Fernando Bolín.
DIRECTOR EDITORIAL REVISTAS DE USUARIOS:
Juan Arencibia.
GERENTE DE CIRCULACION Y VENTAS
Luis Carrero.
PRODUCCION:
Miguel Onieva.
DIRECTOR DE MARKETING:
Antonio González.
SERVICIO AL CLIENTE:
Julia González.
Tel.: 733 79 69
DIRECCION Y REDACCION:
C/ Bravo Murillo, 377 - 5.° A.
Tel.: 733 74 13
28020 Madrid.
PUBLICIDAD Y ADMINISTRACION:
Publicidad en Madrid:
Emilio García.
C/ Bravo Murillo, 377 - 3.° E.
Tel.: 733 96 62-96
Publicidad en Barcelona:
Lidia Cendros.
C/ Pelayo, 12.
Tel.: (93) 301 47 00 Ext. 27-28.
08001 Barcelona.
Depósito Legal: M. 16.755-1985
Impreso en Héroes, S.A.
C/ Torrelara, 8. 28016 Madrid.
Distribuye:
S.G.E.L. Avda. Valdelaparra, s/n.
Alcobendas (Madrid).
DISTRIBUIDORES:
PORTUGAL: VASP Sociedad de Transportes e Distribuidores. LISBOA.
CHILE: CORPRODE SANTIAGO
ARGENTINA: DISTRIBUIDORA INTERCONTINENTAL BUENOS AIRES.
El P.V.P. para Ceuta, Melilla y Canarias, incluido servicio aéreo, será de 310 ptas.

SUSCRIPCIONES
Rogamos dirija toda la correspondencia relacionada con suscripciones a:
MSX
EDISA. Tel. 415 97 12
C/Lopez de Hoyos, 141-5.°
28002 MADRID
(Para todos los pagos reseñar solamente MSX)
Para la compra de ejemplares atrasados diríjanse a la propia editorial
MSX
C/Bravo Murillo, 377-5.° A
Tel. 733 74 13 28020 MADRID

Si deseas colaborar en MSX remite tus artículos o programas a Bravo Murillo 377, 5.° A 28020 Madrid. Los programas deberán estar grabados en cassette y los artículos mecanografiados.
A efectos de remuneración, se analiza cada colaboración aisladamente, estudiando su complejidad y calidad.

CATALOGO DE SOFTWARE PARA ORDENADORES PERSONALES IBM

TODO EL CATALOGO DE SOFTWARE CON MAS DE 800 FICHAS





1.ª ENTREGA 3.500,— PTAS.
(400 FICHAS + FICHERO)

RESTO EN TRES ENTREGAS TRIMESTRALES DE 1.500,— PTAS. CADA UNA.

PRECIO TOTAL DE LA SUSCRIPCION - 8.000,— PTAS.

CUPON DE PEDIDO

SOLICITE **HOY MISMO**
EL CATALOGO DIRECTAMENTE A

infodis, s.a.

BRAVO MURILLO, 377 - 5.° A
28020 MADRID

O EN LOS CONCESIONARIOS IBM

El importe lo abonaré: POR CHEQUE ☐ CONTRA REEMBOLSO ☐
CON MI TARJETA DE CREDITO ☐ Ref: CATALOGO DE SOFTWARE

Cargue **8.000** ptas. a mi tarjeta American Express ☐ Visa ☐ Interbank

Número de mi tarjeta ______

Fecha de caducidad ______ Firma

NOMBRE ______

CALLE ______

CIUDAD ______ D.P. ______

PROVINCIA ______

MSX

DYNADATA

NO LO DUDE

PRECIO INSUPERABLE



TARJETA PROGRAMA.
EL NUEVO FORMATO DEL FUTURO.

MONITOR 12″
FOSFORO
VERDE

ADAPTADOR TARJETA

DATA CASSETTE

NUEVO

DYNADATA MSX
CON TECLADO EN



* Letras y signos iguales al teclado del PC de IBM.

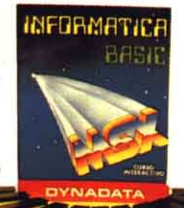
Ordenador DYNADATA MSX con teclado en español	46.900
DYNADATA MSX con cassette y monitor de color	95.500
DYNADATA MSX y unidad de diskette de 5 ¼″ de 360 Kbytes con monitor de fósforo verde	108.900
con monitor de color	141.000
TARJETA/PROGRAMA del juego LE MANS con adaptador, el cual sirve para cualquier tarjeta que usted adquiera. Precio especial con la compra del DYNADATA MSX	4.900

Con el DYNADATA MSX usted podrá:

- Divertirse con la amplia gama de juegos MSX.
- Aprender Informática y Basic con el curso autodidáctico y audiovisual.
- Llevar gestiones administrativas con los programas de proceso de textos, base de datos, contabilidad, stock, recibos, etc.
- Ayudar a sus hijos en sus estudios de 3.º a 8.º de EGB con los programas de Matemáticas, Lenguaje y Ciencias Naturales.
- Aprender idiomas tan necesarios como el inglés.
- Programar con los lenguajes: LOGO, PASCAL, FORTRAN y COBOL.

CURSO DE INFORMATICA Y BASIC

- Autodidáctico
- Audiovisual
- 12 cassettes
- 24 lecciones
- Evaluaciones periódicas
- Diploma Fin de Curso



Especificaciones DYNADATA MSX: Procesador Z80A, 64 Kbyte RAM, 16 Kbyte VRAM, 32 Kbyte ROM, 24 líneas x 40 columnas, 256 x 192 pixels, 16 colores, MSX-BASIC, MSX-DOS.

DYNADATA

Y:

SONY	PIONEER
PHILIPS	YAMAHA
CANON	MITSUBISHI
SANYO	GOLDSTAR
JVC	SAMSUNG
TOSHIBA	HITACHI
SPECTRAVIDEO	MATSUSHITA
	CASIO

Se han decidido por MSX. Esto le permite compartir los programas y periféricos con todas estas reconocidas marcas.

Por todo, NO LO DUDE. Decídase por



Solicite información: Sor Angela de la Cruz, 24 - 28020 Madrid. Tels. (91) 279 21 85 - 279 28 01 - 270 01 93. Telex 44619 DYNA
Delegación Barcelona: Aribau, 61, entlo - 08011 Barcelona. Tels. (93) 254 73 04 - 254 73 03